

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-128349

(43)Date of publication of application : 16.05.1997

(51)Int.Cl.

G06F 15/16

G06F 15/16

G06F 11/18

(21)Application number : 08-145550

(71)Applicant : TANDEM COMPUT INC

(22)Date of filing : 07.06.1996

(72)Inventor : ROBERT W HORST
GARCIA DAVID J

(30)Priority

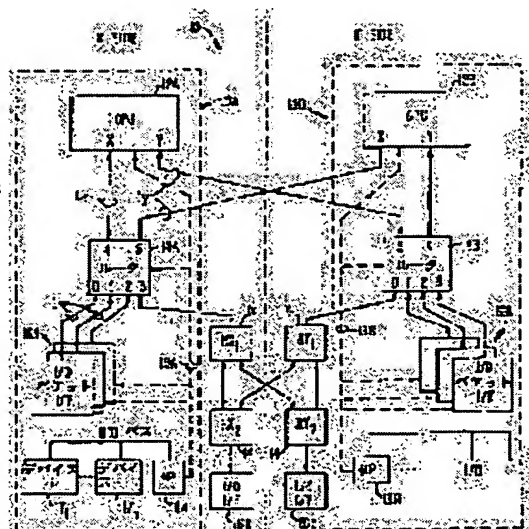
Priority number : 95 485062 Priority date : 07.06.1995 Priority country : US

(54) FAIL-FIRST, FAIL-FUNCTIONAL, AND FAULT-TOLERANT-MULTIPROCESSOR-SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To facilitate fault-tolerant operation by providing a network which mutually connects a central processor and an input/output device so that one of central processors gains communication access to one of input/output devices without requesting other's use.

SOLUTION: The MPs 18 of subprocessor systems 10A and 10B connect an IEEE1149. one-test bus 17 registers used by the MPs 18 to elements of the subprocessor systems thorough on-line access port interfaces included in the elements so as to transmit states and control information between the elements and MPs 18. The MPs 18 generate and send message packets to communicate with a CPU 12. The CPU 12, a router 14, and an I/O packet interface 16 are mutually connected by a TNet link L and have a two-way data communication.



LEGAL STATUS

[Date of request for examination]

06.06.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-128349

(43) 公開日 平成9年(1997)5月16日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/16	3 6 0		G 0 6 F 15/16	3 6 0 R
	4 7 0			4 7 0 J
11/18	3 1 0		11/18	3 1 0 A

審査請求 未請求 請求項の数1 OL (全 84 頁)

(21) 出願番号 特願平8-145550

(22) 出願日 平成8年(1996)6月7日

(31) 優先権主張番号 08/485062

(32) 優先日 1995年6月7日

(33) 優先権主張国 米国 (US)

(71) 出願人 391058071

タンデム コンピューターズ インコーポ
レイテッドTANDEM COMPUTERS IN
CORPORATED

アメリカ合衆国 カリフォルニア州

95014 クーパーティノ ノース タンタ
ウ アベニュー 10435

(72) 発明者 ロバート ダブリュー ホースト

アメリカ合衆国 カリフォルニア州

95070 サラトガ ラーチモント アベニ
ュー 12386

(74) 代理人 弁理士 中村 稔 (外6名)

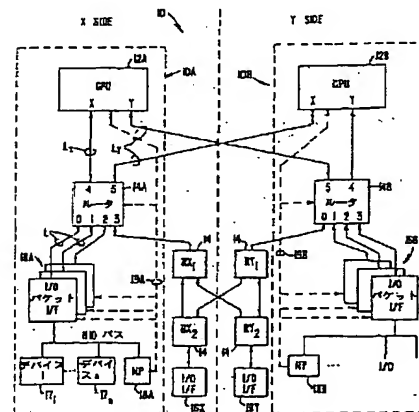
最終頁に続く

(54) 【発明の名称】 フェイラーファースト、フェイラーファンクショナル、フォルトトレラント・マルチプロセッサ・システム

(57) 【要約】 (修正有)

【課題】 フェイラーファースト、フェイラーファンクショナル、フォルトトレラント・マルチプロセッサ・システムを提供する。

【解決手段】 マルチプロセッサ・システムは、それぞれが実質的に同一に構成された多数のサブプロセッサ・システムを含む。サブプロセッサ・システムの一つのCPUは、システムのI/O装置、またはシステムのCPUと、ルーティング装置を通して、通信しうる。I/O装置とCPUsとの間の通信は、パケット化されたメッセージによって行なわれる。CPUs及びI/O装置は、システムのCPUのメモリへ書込まれるか、またはそれから読取られる。メモリ保護は、そのCPUのメモリへの読取り/書込みに対する妥当性検査を備えているような各CPUによって保守される。



【特許請求の範囲】

【請求項1】 複数の中央処理装置と、複数の入力／出力装置と、前記複数の中央処理装置のいずれか他のものの使用を要求することなく前記中央処理装置のいずれか一つが前記入力／出力装置のいずれか一つへの通信アクセスを有するように前記中央処理装置及び前記入力／出力装置を相互接続するネットワークとを備えていることを特徴とする多重処理システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、一般にデータ処理システムに関し、特に多重処理システム並びにプロセッサ間及び入力／出力連絡の連結性を供給するリライアブル・システム・エリア・ネットワークに関する。更に、システムは、フォルトトレラント機能を示すべく構成される。

【0002】

【従来の技術】今日のフォルトトレラント・コンピューティングは、特殊な軍事及び通信システムから汎用の高可用性(high availability)商用(commercial)システムに発展した。フォルトトレラント・コンピュータの発展は、よく記録されている(D. P. Siewiorek, R. S. Swarz, "The Theory and Practice of Reliable System Design", Digital Press, 1982及びA. Avizienis, H. Kopetz, J. C. Laprie, eds, "The Evolution of Fault Tolerant Computing", Vienna: Springer-Verlag, 1987を参照)。初期の高可用性システムは、軍事アプリケーションに対してアイ・ビー・エム(IBM)、ユニヴァック(Univac)、及びレミントン・ランド(Remington Rand)によって1950年代に開発された。1960年代には、アメリカ航空宇宙局(NASA)アイ・ビー・エム(IBM)、エス・アール・アイ(SRI)、シー・エス・ドレイパー研究所(C. S. Draper Laboratory)及びジェット推進研究所(Jet Propulsion Laboratory)は、航空宇宙アプリケーションに対するガイダンスコンピュータの開発にフォルトトレラントを適用し始めた。また、1960年代には、最初の米国電信電話会社(AT&T)製電子スイッチングシステムの開発もあった。

【0003】最初の商用フォルトトレラント・マシンは、オンライン・トランザクション処理アプリケーション用に1970年代にタンデム・コンピュータによって紹介された(J. Bartlett, "A NonStop Kernel", in proc. Eighth Symposium on Operating System Principles, pp. 22-29, Dec. 1981)。1980年代には、多数他の商用フォルトトレラント・システムが紹介された(O. Serlin, "Fault-Tolerant Systems in Commercial Applications", Computer, pp. 19-30, August, 1984)。現行の商用フォルトトレラント・システムは、分散型メモリ・マルチプロセッサ、共用メモリ・トランザクション・ベース・システム、"ペア・アンド・スぺア(pair-and-sp

are)"ハードウェア・フォルトトレラント・システム(R. Freiburghouse, "Making Processing Fail-safe", Mini-micro Systems, pp. 255-264, May 1982;米国特許第 4,907,228号公報もこのペア・アンド・スぺア、及び共用メモリ・トランザクション・ベース・システム技術の一例である)、及びこの出願及びその中に開示された発明の出願人である、アメリカ合衆国カリフォルニア州カッパティノのタンデム・コンピュータ・インコーポレイティッドによって製造される"インテグリティ(Integrity)"コンピューティング・システムのようなトリプルモジュラーリダンダント(triple-modular-redundant)・システムを含む。

【0004】商用フォルトトレラント・コンピュータの殆どどのアプリケーションは、オンライン・トランザクション処理に分類される。金融機関は、電子資金移動(electronic funds transfer)、自動窓口機(ATM)の制御、及び株式市場取引システム(stock market trading systems)に対して高可用性を必要とする。製造メーカーは、自動化された工場の制御、在庫管理、及びオンライン文書アクセス・システムに対してフォルトトレラント・マシンを用いる。フォルトトレラント・マシンの他のアプリケーションは、予約システム、行政データベース、賭け事(wagering)システム、及び電気通信システムを含む。

【0005】フォルトトレラント・マシンのベンダー(Vendors)は、増大したシステム可用性、連続処理の両方、及び故障が存在するときでもデータの正確性を達成することを試みる。特定のシステム・アーキテクチャにより、故障にもかかわらずランを継続するか、または処理が故障が起きたときに最も近いチェックポイントから自動的に再開されるかのいずれかである。あるフォルトトレラント・システムは、故障したコンポーネントについて再構成可能であるべく十分なコンポーネント冗長性を備えているが、故障したモジュールで走行している処理は、失われる。商用フォルトトレラント・システムのベンダーは、プロセッサ及びディスクを越えてフォルトトレランスを拡張した。信頼性における大きな改良を行うために、電源、ファン及びモジュール間接続を含んでいる、故障の全てのソース(原因)がアドレスされなければならない。タンデム・コンピュータ・インコーポレイティッドによって製造される"ノン・ストップ(Non Stop)"及び"インテグリティ"アーキテクチャ(この出願の出願人による、両者がそれぞれ広範に説明された米国特許第 4,228,496号公報、米国特許第 5,146,589号公報及び米国特許第 4,965,717号公報;NonStop(ノン・ストップ)及びIntegrity(インテグリティ)は、タンデム・コンピュータ・インコーポレイティッドの登録商標である)は、商用フォルトトレラント・コンピューティングに対する二つの現行のアプローチを表わす。上記米国特許第 4,278,496号公報に一般に示した、ノン・

ストップ・システムは、一つのハードウェア・コンポーネントの故障にも係わらず動作（オペレーション）を継続すべく設計されたマルチプル・プロセッサ・システムを用いるアーキテクチャを採り入れている。通常動作では、各プロセッサ・システムは、“ホット・バックアップ”としてよりも、その主要コンポーネントを独立にかつ同時に用いる。ノン・ストップ・システム・アーキテクチャは、プロセッサ間連絡に対してバスによって相互接続された16個のプロセッサ・システムまで含む。各プロセッサ・システムは、メッセージ・ベース・オペレーティング・システムのコピーを含むそれ自身のメモリを有する。各プロセッサ・システムは、一つ以上の入力／出力（I/O）バスを制御する。I/Oコントローラ及び装置の双方向ポーティング（dual-porting）は、各装置への多重経路を供給する。ディスク記憶装置のような、（プロセッサ・システムに対する）外部記憶装置は、冗長永久データ記憶（redundant permanent data storage）を維持すべくミラーされうる。

【0006】このアーキテクチャは、“フェイルーフアースト（fail-fast）”動作を供給すべく各システム・モジュールに自己検査ハードウェアを供給する：動作は、他のモジュールの汚染を防ぐためにフォルトが発生したならば一時停止される。フォルトは、例えば、パリティ検査、重合（複写）及び比較、並びに誤り検出コードによって検出される。フォルト検出は、主にハードウェアの責任であり、フォルト回復は、ソフトウェアの責任である。また、ノン・ストップ・マルチプロセッサ・アーキテクチャでは、アプリケーション・ソフトウェア（“プロセス”）は、プライマリ・プロセス及びバックアップ・プロセスを含んでいる、“プロセスペア”としてオペレーティング・システム下でシステム上で走りうる。プライマリ・プロセスは、一つの多重プロセッサ上で走り、バックアップ・プロセスは、異なるプロセッサ上で走る。バックアップ・プロセスは、通常、活動停止状態であるが、プライマリ・プロセスからのチェックポイント・メッセージに応じてその状態を周期的に更新する。チェックポイント・メッセージの内容は、完全状態更新の形、または先のチェックポイント・メッセージからの変化だけを連絡する形を取ることができる。最初は、チェックポイントは、アプリケーション・プログラムに手動で挿入されたが、現行では、ほとんどのアプリケーション・コードは、チェックポイントとトランザクション2段階コミット（transaction two-phase commit）プロトコルとの組合せを通して回復を供給するトランザクション処理ソフトウェアの下で走る。

【0007】タンデム・ノン・ストップ・アーキテクチャのプロセッサ間メッセージ・トラフィックは、各プロセッサが、それ自身を含んでいる、システムの全てのプロセッサによる受信のために“私は、生きている（I'm Alive）”メッセージを周期的にブロードキャストするこ

とを含み、ブロードキャストしているプロセッサがまだ機能していることを他のプロセッサに知らせる。プロセッサが故障したときには、その故障が発表されかつ故障したプロセッサの周期的な“私は、生きている（I'm Alive）”メッセージの欠如によって識別される。応答として、オペレーティング・システムは、最後のチェックポイントからプライマリ・エグゼキューション（primary execution）を始めるべく適切なバックアップ処理を導く。新しいバックアップ処理は、別のプロセッサで開始されうるか、または処理は、ハードウェアが修復されるまでバックアップなしで走らせる。米国特許第 4,817,091号公報は、この技術の一例である。

【0008】各I/Oコントローラは、それが取り付けられた2つのプロセッサの一つによって管理される。コントローラの管理は、プロセッサ間で周期的に切替えられる。管理しているプロセッサが故障したならば、コントローラの所有権は、他のプロセッサへ自動的に切替えられる。コントローラが故障したならば、データへのアクセスは、別のコントローラを通して維持される。ハードウェア・フォルトトレランスを供給することに加えて、上述したアーキテクチャのプロセッサ・ペアは、ソフトウェア・フォルトトレランスのある測定を供給する。プロセッサがソフトウェア・エラーにより故障するときには、バックアップ・プロセッサは、同じエラーに遭遇することなく処理をしばしば成功裏に継続することができる。バックアップ・プロセッサのソフトウェア環境は、異なるキュー・レングス（queue lengths）、テーブル・サイズ、及びプロセス・ミックスを一般に有する。ソフトウェア品質確認検査を逃れてくるソフトウェア・バグのほとんどが、稀なデータ依存境界条件（infrequent data dependent boundary conditions）を含むので、バックアップ処理は、しばしば成功する。上述したアーキテクチャとは対照的に、インテグリティ・システムは、フォルトトレラント・コンピューティングに対する別のアプローチを示す。1990年に紹介された、インテグリティは、ユニックス（Unix）の標準版を走らせるべく設計された（“Unix（ユニックス）”は、アメリカ合衆国デラウェア州のユニックス・システム研究所Inc.の登録商標である）。互換性が主な目的であるシステムでは、ソフトウェアに対する変更をあまり必要としないので、ハードウェア故障回復は、論理的選択である。プロセッサ及び局所メモリは、トリプルモジュラーリダンダンシー（TMR）を用いて構成される。全てのプロセッサは、同じコードストリームを走らせるが、各モジュールの刻時（clocking）は、刻時回路における故障（フォルト）のトレランスを供給すべく独立である。3つのストリームのエグゼキューション（実行）は、非同期であり、かついくつものクロック周期だけ離れてドリフトしうる。ストリームは、グローバル・メモリのアクセスの間中に周期的に再同期される。TMRコントローラ・

ボード上の有権者(voters)は、プロセッサ・モジュールの故障を検出しかつマスクする。メモリは、三重(triplicated) プロセッサ・ボード上の局所メモリと二重(duplicated) TMR Cボード上のグローバル・メモリに区分される。システムの二重部分は、故障を検出するために自己検査技術を用いる。各グローバル・メモリは、双方向ポートされかつプロセッサ並びにI/Oプロセッサ

(IOPs)にインターフェイスされる。標準VME周辺コントローラは、バス・インターフェイス・モジュール(Bus Interface Module) (BIM)を通して一対のバスにインターフェイスされる。IOPが故障であるならば、ソフトウェアは、全てのコントローラの制御を残存するIOPに切替えるためにBIMsを用いることができる。ミラー型ディスク記憶装置は、二つの異なるVMEコントローラに取り付けられうる。

【0009】インテグリティ・システムでは、全てのハードウェア故障は、冗長ハードウェアによってマスクされる。修理後、コンポーネントは、オンラインに再統合される。

【0010】

【発明が解決しようとする課題】先の例は、フォルトトレランスをデータ処理システムに組み込んでいる現在のアプローチを示す。ソフトウェア回復を含んでいるアプローチは、少ない冗長ハードウェアを必要とし、かつあるソフトウェア・フォルトトレランスの可能性を与える。ハードウェア・アプローチは、標準オペレーティング・システムを有する完全互換性を許容しかつ他のシステム上で開発されたアプリケーションを透過的に走らせるべくエクストラ・ハードウェア冗長性を用いる。それゆえに、上述したシステムは、ハードウェア(例えば、冗長性を採り入れている、フェイルーフアンクショナル)またはソフトウェア技術(例えば、高データ・インテグリティ・ハードウェアを有するソフトウェア回復を採り入れている、フェイルーフアースト)のいずれかによりフォルトトレラント・データ処理を供給する。しかしながら、上述したいずれのシステムも、単一データ処理システムによる、ハードウェア(フェイルーフアンクショナル)及びソフトウェア(フェイルーフアースト)アプローチの両方を用いる、フォルトトレラント・データ処理を供給するように構成されているとは思われない。

【0011】上述したような、コンピューティング・システムは、電子データ交換(EDI)及びグローバル・メッセージングのような、電子商業に対してしばしば用いられる。しかしながら、そのような電子商業における今日の要求は、ユーザの数が増加しかつメッセージが更に複雑になるので、更に多くのスループット能力を要求している。例えば、インターネットの最も広く用いられるファシリティである、テキスト・オンリー・イー・メール(e-mail)は、毎年かなり成長している。インタ

ーネットは、画像、音声、及びビデオ・ファイルを送付するためにますます用いられている。音声記憶及び前送り(voice store-and-forward)メッセージは、いたるところにあるようになり、かつデスクトップ・ビデオ会議及びビデオメッセージは、ある組織(organization)で受け入れを取得している。メッセージの各型は、更なるスループットを継続的に要求する。そのような環境では、ローカル・エリア・ネットワーク(LANs)等のような種々の通信ネットワークにより相互接続された、並列アーキテクチャが用いられる。

【0012】サーバ・アーキテクチャに対する重要な要求事項は、膨大な量のデータを移動する能力である。サーバは、データボリュームが増加しかつトランザクションが更に複雑になるときに付加されたスループット能力を加えることができるように、スケラブルな高帯域幅を有すべきである。バス・アーキテクチャは、各システム・コンポーネントに対して利用可能な帯域幅の量を制限する。バス上のコンポーネントの数が増加すると、それぞれに対して更に少ない帯域幅が利用可能である。加えて、即応答は、全てのアプリケーションに対して有利でありかつ対話形アプリケーションに対して必要である。それは、ソース(発生源)からデスティネーション(宛先)にデータを移動するのにどのくらいかかるかの測定である、非常に少ない待ち時間(latency)を必要とする。応答時間に密接に関連した、待ち時間は、サービス・レベル及び被雇用者生産性に影響を及ぼす。本発明の目的は、単一システムで、フォルトトレラント・アーキテクチャ、ハードウェア冗長性及びソフトウェア回復技術に上記二つのアプローチの両方を組み合わせる多重プロセッサ・システムを提供することである。

【0013】

【課題を解決するための手段及びその作用並びに効果】一般に、本発明は、多重サブ(副)処理システムから構成された処理システムを含む。各サブ処理システムは、主要処理素子として、同時に命令ストリームの各命令を実行すべくロックステップ、同期方式で動作している一対のプロセッサを含む中央処理装置(CPU)を有する。サブ処理システムのそれぞれは、CPU及びサブ処理システムのアソート(assorted)された周辺装置(例えば、マス記憶装置、プリンタ等)を含んでいる、より大きな処理システムの種々のコンポーネント間、並びにより大きな総括処理システムを構成しうるサブプロセッサ間に冗長通信経路を供給する入力/出力(I/O)システム・エリア・ネットワーク・システムを更に含む。処理システムのコンポーネント間(例えば、それがどのサブ処理システムに属しうるかに関係なく、CPUと別のCPU、またはCPUといずれかの周辺装置)の通信は、複数の相互接続しているリンクのバス構造(ここでは、“TNet”と称する)によって相互接続される多数のルータ素子を含んでいるシステム・エリア・ネット

ワーク構造により送信またはソース・コンポーネント（例えば、CPU）からデスティネーション（宛先）素子（例えば、周辺装置）に送られるパケット化されたメッセージを形成しかつ送信することによって実施される。ルータ素子は、メッセージ・パケットに含まれた情報に基づいて処理システムの送信コンポーネントから宛先コンポーネントまで適切または利用可能な通信経路を選択する役割を果す。それゆえに、ルータ素子のルーティング能力（機能）は、周辺装置への通信経路をCPUのI/Oシステムに供給するが、プロセッサ間連絡に用いられることをそれに許容もする。

【0014】上記したように、本発明の処理システムは、“フェイルーフースト”及び“フェイルーフアンクショナル”動作の両方を通してフォルトトレラント動作を供給すべく構成される。フェイルーフースト動作は、誤り検査機能をシステムの戦略地点に位置決めすることによって達成される。例えば、各CPUは、CPUの（ロックステップ動作型）プロセッサ素子とその関連メモリとの間の種々のデータ経路における多様な地点で誤り検査機能を有する。特に、本発明の処理システムは、インターフェイスで、かつ性能にほとんど影響を及ぼさないように、誤り検査を実行する。従来技術のシステムは、プロセッサのペアを走らせて、かつデータとプロセッサ及びキャッシュメモリ間の命令フローとを検査（比較）することによって誤り検査を一般に実施する。この誤り検査の技術は、アクセスに遅れを付加する傾向があった。また、この型の誤り検査は、利用可能でありうるオフザシェルフ(off-the-shelf)部分（即ち、単一半導体チップまたはモジュール上のプロセッサ/キャッシュメモリ組合せ）の使用を排除した。本発明は、主メモリ及びプロセッサ・キャッシュ・インターフェイスよりも遅いスピードで動作するI/Oインターフェイスのような、より遅い速度で動作する地点でプロセッサの誤り検査を実行する。更に、誤り検査は、プロセッサ、それらのキャッシュメモリ、I/O及びメモリ・インターフェイスで発生しうる誤りの検出を許容する位置で実行される。これは、それらがパリティまたは他のデータ・インテグリティ検査を必要としないのでメモリ及びI/Oインターフェイスに対するより簡単な設計を許容する。

【0015】処理システムのコンポーネント間の通信フローの誤り検査は、システムの素子間に送られるメッセージ・パケットに巡回冗長検査(CRC)を付加することによって達成される。各メッセージ・パケットのCRCは、メッセージの宛先においてだけでなく、そのソース（発生源）からデスティネーション（宛先）にメッセージ・パケットを送るために用いられる各ルータ素子により宛先への途中でも検査される。メッセージ・パケットが間違ったCRCを有すべくルータ素子によって見出されたならば、メッセージ・パケットは、そのようにタ

グ付けされ、かつ保守診断システムに報告される。この特徴は、故障分離に対する有用なツールを供給する。このようなCRCの使用は、メッセージ・パケットが通過するときにルータ素子がCRCを変更または再生しないので終端間でメッセージ・パケットを保護すべく動作する。各メッセージ・パケットのCRCは、各ルータ・クロッシング(router crossing)で検査される。指令記号—“このパケットは、よい(This packet Good)”(TPG)または“このパケットは、悪い(This packet Bad)”(TPB)—は、全てのパケットに添えられる（付加される）。保守診断プロセッサは、誤りが一時（過渡）的であっても、誤りを導くリンクまたはルータ素子を分離するためにこの情報を用いることができる。

【0016】ルータ素子は、メッセージを受信しかつ送信することができる複数の双方向ポートを備えている。そのように、それらは、多様なトポロジーの用途に適していおり、代替経路を処理システムのいずれかの二つの素子間（例えば、CPUとI/O装置間）に設けることができ、故障が存在する通信に対して、フォルトトレラント・システムをもたらす。更に、ルータ論理は、メッセージ・パケットが受信されるルータ・ポート及びメッセージ・パケットの宛先に基づいて、ある一定のポートを出力としての考慮からディスエーブルする(disabling)機能を含む。そのメッセージ・パケットのルータの出力ポートとしての無許可ポートを示す宛先アドレスを含んでいるメッセージ・パケットを受信するルータは、メッセージ・パケットを捨て、かつ保守診断システムに知らせる。この特徴の適切な使用は、メッセージ・パケットを連続ループ及び遅延に入ることから防ぐかまたは（例えば、以下に説明する、“デッドロック”条件を生成することによって、）そのようにすることから他のメッセージ・パケットを防ぐ。

【0017】処理システムのCPUは、二つの基本モードの一つで動作するように構成される：（ペアの）各CPUが他とは独立に動作する“シンプレックス・モード”か、または、CPUのペアが同期された、ロックステップ方式で動作する“デュプレックス・モード”である。シンプレックス・モード動作は、誤り検査ハードウェア（例えば、各プロセッサが、そのシプリング（同胞）プロセッサの動作可能性を検査し、かつ故障したと思われるかまたは信じられるプロセッサの処理を取って代わる機能を有するようなマルチ処理システムを教示する米国特許第4,228,496号公報）によって検出される故障から回復する機能を供給する。デュプレックス・モードで動作するときには、ペアになったCPUは、両方ともに同様な命令ストリームを実行し、ペアの各CPUは、実質的に同時にストリームの各命令を実行する。

【0018】デュプレックス・モード動作は、あまり堅牢でない(less robust)オペレーティング・システム

(例えば、ユニックス (UNIX) オペレーティング・システム) に対するフォルトトレラント・プラットフォームを供給する。ペアになった、ロックステップCPUを有する、本発明の処理システムは、故障が、主にハードウェアを通して、多くの場合にマスクされる (即ち、故障の存在にも係わらず動作する) ように構成される。処理システムがデュプレックス・モードで動作しているときには、各CPUペアは、周辺装置が表面上どの (二つ以上の) サブプロセッサ・システムのメンバーであるかに係わりなく、処理システムの周辺装置にアクセスするためにI/Oシステムを用いる。また、デュプレックス・モードでは、CPUペアへの送付向けメッセージ・パケットは、CPUペアの同期、ロックステップ動作を維持するために実質的に同時にI/Oシステムによりペアの両方のCPUに送付される。それゆえに、本発明の主な創作的態様は、ロックステップ・ペアの両方のCPUが同じ方法で同じ時間にI/Oメッセージ・パケットを受信することを確実にする機能を有するデュプレックスの動作モードを供給する。これに関しては、デュプレックス・ペアの一つのCPUに接続されたルータ素子は、ペアの両方のCPU素子に接続される。(マス記憶装置のような周辺装置かまたは処理装置から) CPUペアに対してメッセージを受信することによって、そのように接続された、ルータは、メッセージを複写しかつCPUが同期されたままであることを確実にする同期方法を用いてそれをペアの両方のCPUに送付する。結果として、I/Oシステム及び他のデュプレックスCPUペアから見た、デュプレックスCPUペアは、単一CPUとして見られる。それゆえに、全てのサブ処理システムからの素子を含む、I/Oシステムは、周辺装置がアクセス可能であるような一つのホモジニアス(homogeneous) システムとしてデュプレックスCPUペアによって見られるようにされる。

【0019】本発明の別の重要かつ新規な特徴は、ルータ素子の多用 (融通) 性がいずれかのCPUが実際には一対の同期された、ロックステップCPUであるようなマルチプロセッサ・システムを形成すべくデュプレックス・モード・オペレーティング・システム・ペアのクラスタを結合させるということである。本発明の更に別の重要な態様は、I/O素子から発行されている割込みが、他の情報転送と同じ方法で、即ち、メッセージ・パケットによって、CPU (またはデュプレックス・モードの場合にはCPUペア) に伝達されるということである。これは、多数の利点を有する： 割込みは、通常のI/Oメッセージ・パケットのように、CRCによって保護することができる。また、両方のCPUへの同時送付に対する信号への割込み専用の追加信号回線の要求が不要になる；メッセージ・パケット・システムを介して割込みを送付することは、それらが、I/Oメッセージ・パケットと同じ方法で、同期されたファッショ

でデュプレックスされたCPUに到着することを確実にする。割込みメッセージ・パケットは、割込みの原因となる情報を含み、現在行われるように、CPU(s) が原因を決定すべく割込みを発行している装置を読取るという時間のかかる要求を不要にする。更に、上記したように、ルーティング素子は、割込みパケット送付に対する多重経路を供給することができ、それによってシステムのフォルトトレラント能力を上昇させる。加えて、I/O装置とCPUの間でデータを伝達しかつCPUに割込みを伝達するために同じメッセージング・システムを用いることは、I/O及び割込みの順番を維持する；即ち、I/O装置は、割込みメッセージが送られる前にI/Oが終了するまで待つ。

【0020】本発明の更なる新規な態様は、CPUのメモリへのアクセスの妥当性を検査する技術の実施である。本発明により構成されたような、処理システムは、CPUのメモリをシステムの他の素子 (即ち、他のCPU及び周辺装置) によってアクセスさせる。そうであるならば、不注意な及び/又は無許可なアクセスに対して保護する方法が供給されなければならない。本発明のこの態様に従って、各CPUは、そのCPUのメモリへのアクセスが許可されるCPUの外部の各ソースに対するエントリを含んでいるアクセス妥当性検査及び変換 (AVT) 表を維持する。各そのようなAVT表エントリは、許可されたアクセスの型 (例えば、メモリへの書込み)、及びそのアクセスが許可されるメモリの場所に対するような情報を含む。I/Oシステムを通して送られるメッセージ・パケットは、メッセージ・パケットのオリジネータ(originator)、メッセージ・パケットの宛先、メッセージが含んでいるもの (例えば、宛先で書込まれるべきデータ、または宛先から読取られるべきデータに対する要求)、等を記述している情報を有して、上述したように、生成される。ルータ素子にその最終宛先にメッセージ・パケットを迅速に送らせることに加えて、受信CPUは、メッセージ・パケットのソースに関係している (属している) エントリに対してAVT表をアクセスし、アクセスが許可されるかどうか、かつそうであるならば、受信CPUがリマップ (即ち、変換) すべく選択するアドレスの型及びその場所を理解するために検査すべく情報を用いる。この方法で、CPUのメモリは、逸脱したアクセスに対して保護される。また、AVT表は、CPUへ割込みを通過するためにも用いられる。

【0021】AVT表は、CPUのメモリが故障のI/O装置によって汚染されないことを保証する。アクセス権は、1バイトからページの広がりまでの大きさの範囲であるフォーム・メモリ(form memory) に付与することができる。システムのシステム・ベンダーは、サードパーティー周辺サプライヤのハードウェア及びソフトウェアの品質に関してより少ない制御を通常有するので、

この故障封じ込め(fault containment)は、I/Oにおいて特に重要である。問題は、I/Oシステム全体よりも単一のI/O装置またはコントローラに分離することができる。本発明の更なる態様は、データをI/Oに送信するためにCPUによって用いられる技術を含む。本発明のこの態様によれば、CPUとプロセッサ・システムの他のコンポーネントとの間の入力/出力情報転送を処理するために各CPUにブロック転送エンジンが備えられる。それによって、CPUの個別のプロセッサ装置は、メモリからTNetネットワーク上に情報を得ること、またはネットワークから情報を受け入れることのよりありふれたタスク(more mundane tasks)から取り除かれる。CPUのプロセッサ装置は、所望の宛先、データの量、及び応答が必要であるならば、受信したときに応答が配置されるメモリの場所、のような他の情報を伴った、送られるべきデータを含んでいるメモリにデータ構造を単にセットアップする。プロセッサ装置がデータ構造を生成するタスクを終了したときには、ブロック転送エンジンは、それが取って代わり、かつメッセージ・パケットの形で、データを送ることを始める原因となることが知られる。応答が期待されるならば、ブロック転送エンジンは、応答が行くメモリの場所を含んでいる、応答を処理するために必要な構造をセットアップする。応答が受信されるとき及び受信されたならば、それは、識別された期待メモリ位置に送られ、かつ応答を受信したことをプロセッサ装置に知らせる。

【0022】本発明の更なる態様及び特徴は、添付した図面に関連して本発明の以下の詳細な説明を読むことにより当業者に明らかになるであろう。

【0023】

【実施例】

概要：ここで、図1を参照すると、本発明の種々の教示により構成された、データ処理システムが参照番号10で示されている。図1が示すように、データ処理システム10は、それぞれの構造及び機能が実質的に同じである二つのサブプロセッサシステム10A及び10Bを備えている。従って、特に示さない限り、サブプロセッサシステム10のいずれ一つの説明は、他のサブプロセッサシステム10に同様に適用されるということが理解されるべきである。従って、図1を続いて参照すると、サブプロセッサシステム10A、10Bのそれぞれは、中央処理装置(CPU)12、ルータ14、及びそれぞれがネイティブ(固有の)入力/出力(NIO)バスによって多数(n)のI/O装置17に結合される複数の入力/出力(I/O)パケット・インターフェイス16を含んで示されている。I/Oパケット・インターフェイス16の少なくとも一つは、保守プロセッサ(MP)18にも結合される。

【0024】各サブプロセッサシステム10A、10BのMP18は、IEEE1149.1テスト・バス17

(図1では、想像線で示す；明瞭化のために図2及び図3では示されていない)及び、状態及び制御情報を素子とMP18の間に伝達(通信)するためにMP18によって用いられるレジスタを、各素子に対して、含むオンライン・アクセス・ポート(OLAP)インターフェイスを介してそのサブプロセッサシステムの素子のそれぞれに接続する。MP18は、メッセージ・パケットを生成しかつ送ることによって、図1に示すように、CPU12とも通信することができる。(実際には、MP18からの要求に応じてパケットを生成しかつ送るのは、I/Oパケット・インターフェイス16である。)

CPU12、ルータ14、及びI/Oパケット・インターフェイス16は、“TNet”リンクLによって相互接続され、双方向データ通信を供給する。各TNetリンクLは、二つの一方方向10ビット・サブリンク・バスを備えている。各TNetサブリンクは、データの9ビット及び付随クロック信号を運ぶ。図1に更に示すように、TNetリンクLは、サブプロセッサシステム10A及び10Bを互いに相互接続もし、各サブプロセッサシステム10に他の並びにCPU間通信のI/O装置へのアクセスを供給する。理解されるように、処理システム10のCPU12は、そのようなアクセスは確証されなければならないが、他のCPU12のメモリへのアクセスを付与されることができる。本発明の重要な側面である。ある程度類似なファクションで、CPU12のメモリは、通常CPUによって起動された動作の結果として、周辺装置へもアクセス可能である。これらのアクセスは、方向の定まらない(wayward)周辺装置17によるCPU12のメモリの汚染を防ぐためにも確証される。

【0025】サブプロセッサシステム10A/10Bは、図1(及び以下に説明する図2、図3)に示すように一対になるのが好ましく、各サブプロセッサシステム10A/10Bペアは、CPU12、少なくとも一つのルータ14、及び関連I/O装置を有する少なくとも一つのI/Oパケット・インターフェイス16を含んでいる。

【0026】各CPU12は、そこにおいてメッセージ・パケットが送信及び/又は受信される、二つのI/Oポート、Xポート及びYポートを有する。CPU12

(例えばCPU12A)のXポートは、対応サブプロセッサシステム(例えば10A)のルータ(14A)へTNetリンクLによって、接続する。逆に、Yポートは、CPU(12A)をコンパニオン・サブプロセッサシステム(10B)のルータ(14B)に接続する。この後者の接続は、他のサブプロセッサシステム(10B)のI/O装置へのCPU(12A)によるアクセスに対する通信経路だけでなく、CPU間通信に対してそのシステムのCPU(12B)への通信経路も供給する。情報は、メッセージ(パケット)を介して処理シス

テム10の素子とシステムの他の素子（例えば、サブプロセッサシステム10AのCPU12A）及びシステムの他の素子（例えば、サブプロセッサシステム10BのI/Oパケット・インターフェイス16Bに関連したI/O装置）との間で伝達される。各メッセージ・パケットは、データを含みうるかまたは指令記号でありうる多数の9ビット記号で構成される。メッセージ・パケットは、メッセージ・パケットを送信しているコンポーネントによって供給される送信機クロックを伴った、ビット一並列、記号一直列ファッションで、TNetリンクL上に同期的に送信される。通信素子（即ち、送信機及び受信機）間のクロックは、二つのモード、＜近周波数（near frequency）＞モード、または＜周波数封じ込み（frequency locked）＞モードの一つで動作されうる。

【0027】近周波数で動作するときには、送信素子及び受信素子によって用いられるクロック信号は、所定のトレランス（許容範囲）内で一実質的に同じ周波数であるべく制約されるが、別々であり、かつ局所的に生成される。この理由により、クロック同期先入り先出し（CS FIFO）記憶装置構造（以下に詳述）を用いて、受信機で記号を受信する固有の方法が開発された。CS FIFOは、近周波数動作の結果としてメッセージ・パケットの受信機と送信機のクロック信号間に起こりうるスキューを吸収すべく動作する。近周波数動作は、一つのルータ14から別のルータへ、またはルータ14とI/Oパケット・インターフェイス16との間で、またはルータ14とシンプレックス・モード（以下に詳述）で動作しているCPU12の間で記号を送信するとき用いられる。

【0028】周波数封じ込み動作は、同相である必要はないが、送信機及び受信機装置のクロック信号の周波数が封じ込み(lock)られることを意味する。周波数封じ込みクロック信号は、ルータ14A、14Bと対になったサブプロセッサシステム（例えば、図1のサブプロセッサシステム10A、10B）のCPU12との間で記号を送信するために用いられる。送信及び受信素子のクロックは、位相関係にないもので、近周波数動作に対して用いられるものとは多少異なるモードで動作するにもかかわらずクロック同期FIFO（CS FIFO）が再び用いられる。各ルータ14は、一つを除き、それぞれが実質的に同じように構成される、6個の双方向TNetポート、0～5を備えている。CPU12に接続するために用いられる二つのポート（4、5）は、ある程度異なって構成される。この相違は、理解されるように、サブプロセッサシステム10のペアが、各CPU12が同じ命令ストリームから同時に同じ命令を実行するために動作する、デュプレックス・モードと呼ばれる、同期された、ロックステップ・モードで動作することができるという事実による。デュプレックス・モードでは、ある一つのI/O装置からの入力I/Oは、実質的

に同じ時間に両方のCPU12に供給されるということが重要である。それゆえに、例えば、ルータ14Aのポート3で受信したメッセージ・パケットは、ルータ14Aによって複写（複製）されかつ同じ記号が実質的に同じ時間にCPU12へ伝達されるようにルータ4、5から送信される。ポート4、5がルータ14の他のポート0～3とは異なるのは、この点においてである。

【0029】図1は、本発明の別の特徴である、追加のルータ14（図1では、ルータRX₁、RX₂、RY₁、及びRY₂で示される）の使用による二つのサブプロセッサシステム10A、10B間のクロス・リンク接続を示す。図1に示すように、追加されたルータRX₁、RX₂、RY₁、及びRY₂は、それらをI/Oパケット・インターフェイス16X、16Yに結合するためにサブプロセッサ10A、10B（または、示したように、それぞれX及びY“側”）間にクロス・リンク接続を形成する。ルータRX₁—RY₂及びRY₁—RX₂間のクロス接続リンクは、CPU12A、12Bとルータ14B、14Aの間のクロス・リンク接続Lyと同じように一つの側（XまたはY）から別の側へクロス・リンク経路を供給する。しかしながら、ルータRX₁、RX₂、RY₁、及びRY₂によって供給されるクロス・リンクは、I/Oパケット・インターフェイス16X、16Yに接続されうるI/O装置（図示省略）を一つの側（XまたはY）又は別の側へ送らせる。図1に示すように、ルータRX₂及びRY₂は、I/Oパケット・インターフェイス装置16x及び16yにデュアル・ポートされたインターフェイスを供給する。もちろん、I/Oパケット・インターフェイス16X、16Yが、ルータRX₂及びRY₂によって形成されたデュアル・ポート接続によって供給されたクロス・リンク接続に対する代替としてデュアル・ポート及びルータRX₁、RY₁に接続するためのそれらデュアル・ポートを有すべくそれ自身を構成することができることは、いま明らかである。

【0030】ルータ14の構成及び設計が理解されたときに明らかになるように、それらは、図2及び図3に示すような追加のサブプロセッサシステムを含むべく処理システム10の構造を拡張できるように用いるのに向いている。図2では、例えば、ルータ14A及び14Bのそれぞれの一つのポートは、対応サブプロセッサシステム10A及び10Bを追加のサブプロセッサシステム10A'及び10B'に接続するために用いられ、それにより、図1の基本処理システム10のクラスタを含んでいる更に大きな処理システムを形成する。同様に、図3において上記の内容は、サブプロセッサシステム・ペア10A/10B、10A'/10B'、10A''/10B''、及び10A'''/10B'''を備えている、8つのサブプロセッサシステム・クラスタを形成すべく拡張される。次に、サブプロセッサシステムのそれ

ぞれ（例えば、サブプロセッサシステム10A）は、図3が示すように、サブプロセッサシステム10A及び10Bが、サブプロセッサシステム10A' / 10B' , 10A' ' / 10B' ' , 及び10A' ' ' / 10B' ' ' を越えてクラスタを拡張するために、追加のルータ14C及び14Dをそれぞれ含むということを除き、CPU12、ルータ14、及びI/Oパケット・インターフェイス16によってTNetに接続されたI/Oの同じ基本最小構成を本質的に有する。更に図3に示すように、ルータ14C及び14Dの未使用ポート4及び5は、クラスタを更に拡張するために用いる。

【0031】ルータ14の設計、並びにシステム10のトポロジを構成するときのルータ14の適切な使用と共に、メッセージ・パケットを送るために用いる方法により、図3の処理システム10のCPU12は、他のサブプロセッサシステムの他の“端末装置”（例えば、CPU又は/及びI/O装置）にアクセスできる。二つの経路が、CPU12からI/Oパケット・インターフェイス16に接続している最後のルータ14まで利用可能である。例えば、サブプロセッサシステム10B' のCPU12Bは、（サブプロセッサシステム10B' ）のルータ14B、ルータ14D、及び（サブプロセッサシステム10B' ' ）のルータ14B及び、リンクLAを介して、ルータ14A（サブシステム10A' ' ' ）、（サブシステム10A' の）ルータ14Aを介するOR、ルータ14C、及びルータ14A（サブシステム10A' ' ' ）を介してサブプロセッサシステム10A' ' ' のI/O16' ' ' をアクセスできる。同様に、サブ処理システム10A' ' のCPU12Aは、データを読み取りまたは書き込むためにサブプロセッサ10BのCPU12Bに含まれるメモリを（二つの経路を介して）アクセスしうる。（処理システムの別のコンポーネントの一つのCPU12によりメモリ・アクセスは、分かるように、そのようにするための許可を有することのアクセスを求めているコンポーネントを必要とする。この点において、各CPU12は、そのCPUのメモリをアクセスするための許可を有している各コンポーネントに対するエントリを含み、通常、そのアクセスをメモリの選択されたセクションに制限し、かつ許可したアクセスの型を制限している、表を維持する。このような方法で許可を要求することは、誤ったアクセスによるCPUのメモリ・データの汚染を防ぐ。）

【0032】図2に示す処理システムのトポロジは、サブプロセッサシステム10A' , 10B' のルータ14A' , 14B' に接続すべく、ルータ14A、14Bのポート1、及び補助TNetリンクLAを用いて達成される。それによって得られたトポロジは、図2に示す処理システム10のCPU12（12A、12B、12A' , 12B' ）とI/Oパケット・インターフェイス16との間に冗長通信経路を確立する。例えば、サブ

プロセッサシステム10A' のCPU12A' は、ルータ14A'（インポート4、アウトポート3）、ルータ14A（インポート3、アウトポート0）、及び関連相互接続TNetリンクLによって形成された第1の経路によりサブプロセッサシステム10AのI/O16Aをアクセスしうる。しかしながら、ルータ14A' が失われたならば、CPU12A' は、ルータ14B'（インポート4、アウトポート3）、ルータ14B（インポート3、アウトポート1）、リンクLA、及びルータ14A（インポート1、アウトポート0）によって形成された経路によりI/O16Aをアクセスしうる。また、図2のトポロジは、システム10のCPU12のペア間に冗長通信経路をも確立し、フォルトトレラントCPU間通信に対する手段を供給するという点に注目する。

【0033】図3は、図2に示したもののトポロジの拡張を示す。各サブプロセッサ・ペアの各ルータ14の一つのポートを相互接続し、サブプロセッサシステム10A' ' , 10B' ' 及び10A' ' ' , 10B' ' ' のルータ14（14A' ' 及び14B' ' ' ）のポート1間に追加の補助TNetリンクLA（鎖線接続で図3に示す）を用いることによって、二つの分離、独立データ経路がCPU12とI/Oパケット・インターフェイス16との間に見出すことができる。このファッショでは、端末装置（例えば、CPU12またはI/Oパケット・インターフェイス16）は、他の端末装置への少なくとも二つの経路を有する。二つの端末装置間（例えば、図3のシステム10における、CPU12と他のCPU12の間、またはCPU12とI/Oパケット・インターフェイス16との間）のアクセスの代替経路を供給することは、重要な概念である。フォルト・ドメイン(fault domain)の損失は、二つの残っているフォルト・ドメイン間の通信を分断しない。ここで、フォルト・ドメインは、サブプロセッサシステム（例えば、10A）でありうる。それゆえに、ルータ14A' ' ' 及び14B' ' ' 間に補助TNetリンクLAなしで、電力が供給される故障によりサブプロセッサシステム10Aが破壊されたならば、サブプロセッサシステム10BのCPU12Bは、（ルータ14A、ルータ14C、ルータ14A' ' ' を介して、I/Oパケット・インターフェイス16' ' ' へ）I/Oパケット・インターフェイス16' ' ' へのアクセスを失うであろう。ルータ14A' ' ' 及び14B' ' ' 間の補助接続LAで、サブプロセッサシステム10Aの損失のよるルータ14A（及びルータ14C）の損失を伴っても、CPU12B間の通信は、ルータ14Bのルート、ルータ14D、ルータ14B' ' ' , ルータ14A' ' ' へ、及び最後にI/Oパケット・インターフェイス16' ' ' への補助接続LAを介してまだ可能である。

【0034】CPUアーキテクチャ：ここで図4を参照

すると、CPU12Aがより詳細に示されている。CPU12A及び12Bの両方は、構成及び機能が実質的に同じなので、CPU12Aの詳細だけを説明する。しかしながら、特に示さない限り、CPU12Aの説明がCPU12Bにも同様に適用するということが理解されるであろう。図4に示すように、CPU12Aは、両方のプロセッサ装置20a、20bが同じ命令を受信しかつ実行し、そして時間において実質的に同じ瞬間に、同じデータ及び指令出力を発行するように、同期した、ロックステップ動作に対して構成される一対のプロセッサ装置20a、20bを含む。プロセッサ装置20a、20bのそれぞれは、対応キャッシュメモリ22に、バス21(21a、21b)によって、接続される。用いられる特定の型のプロセッサ装置は、キャッシュメモリ22が必要でないように十分な内部キャッシュメモリを含むことができる。代替的に、キャッシュメモリ22は、プロセッサ装置20の内部でありうるキャッシュメモリを補うために用いることができる。とにかく、キャッシュメモリ22が用いられたならば、バス21は、128ビットのデータ、16ビットの誤り訂正コード(ECC)チェックビットを導通すべく構成され、データ、(データ及び対応ECCに対する)25のタグ・ビット、タグ・ビットをカバーしている3つのチェックビット、22のアドレス・ビット、アドレスをカバーしている3ビットのパリティ、及び7つの制御ビットを保護する。

【0035】また、プロセッサ20a、20bは、X及びYインターフェイス装置24a、24bに分離64ビット・アドレス/データ・バス23を介して、それぞれ結合される。所望ならば、各バス23a、23b上で伝達されたアドレス/データは、これはバスの幅を増大するが、パリティによっても保護することができる。(プロセッサ20は、アメリカ合衆国カリフォルニア州サンタ・クララのシリコン・グラフィックス、インク(Silicon Graphics, Inc.)のMIPSディビジョンから入手可能なような、RISC R4000型マイクロプロセッサを含むべく構成されるのが好ましい。)

X及びYインターフェイス装置24a、24bは、プロセッサ装置20a、20bと(二つのMCハーフ26a及び26bからなる)メモリ・コントローラ(MC)26及びダイナミック・ランダム・アクセス・メモリ・アレー28を含んでいるCPU12Aのメモリ・システムとの間にデータ及び指令信号を伝達すべく動作する。インターフェイス装置24は、72ビット・アドレス/指令バス25により互にかつMc s 26a及び26bに相互接続する。しかしながら、見て分かるように、(ECCの8ビットを伴った)データの64ビット倍長語(ダブルワード)は、インターフェイス装置24によってメモリ28に書込まれ、一つのインターフェイス装置24は、書込まれる倍長語の一つの語(例えば、上位

32ビット部分)だけを駆動し、他のインターフェイス装置24は、倍長語の他の語(例えば、倍長語の下位32ビット部分)。更に、各書込み動作で、インターフェイス装置24a、24bは、誤りを検査するためにそのインターフェイス装置24によって書込まれないデータ上で他のものによって書込まれたデータでクロスチェック動作を実行する;また、読取り動作では、バス25上に置かれたアドレスが同じ方法でクロス・チェックされる。キャッシュメモリ22並びに(主)メモリ28に書き込まれたデータの両方を保護するために用いられる特定のECCは、通常のものであり、かつ単一ビット誤り訂正、ダブルビット誤り検出を供給する。

【0036】概念的に、各倍長語は、“奇数”及び“偶数”語を含む。Mc s 26の一つは、メモリに奇数語を書込み、他のものは、偶数語を書込む。更に、Mc s 26は、その倍長語に対する8ビット誤り訂正コード(ECC)と一緒に、一度に二つの倍長語を書込む。加えて、ECCチェックビットは、倍長語だけでなく、倍長語が書込まれるメモリ位置のアドレスをカバーするためにも形成される。後者がアクセスされたときには、ECCは、単一ビット誤りを修正し、かつアクセスされた倍長語が、倍長語がそれから記憶された位置のアドレスに対応することを検査すると同時にデータに発生しうる、ダブルビット誤りを検出するために用いられる。CPU12Aのインターフェイス装置24a、24bは、CPU12AのX及びY(I/O)ポートをそれぞれサービスするための回路素子を形成する。それゆえに、Xインターフェイス装置24aは、プロセッサシステム10A(図1)のルータ14Aのポートに双方向TNetリンクLxによって接続し、Yインターフェイス装置24bは、TNetリンクLyによりプロセッサシステム10B(図1)のルータ14Bと同様に接続する。Xインターフェイス装置24aは、ルータ14Aとサブプロセッサシステム10AのCPU12Aとの間の全てのI/Oトラフィックを処理する。同様に、Yインターフェイス装置24bは、CPU12Aとサブプロセッサシステム10Bのルータ14Bとの間の全てのI/Oトラフィックに対して役割を果たす。

【0037】Xインターフェイス装置24aをルータ14Aに接続しているTNetリンクLx(図1、図2及び図3)は、上述したように、それぞれがクロック信号、及び9ビットのデータを運ぶ、二つの10ビット・バス30_x、32_xを含む。バス30_xは、ルータ14Aに送信されたデータを運ぶ;バス32_xは、ルータ14Aから入力してくるデータを運ぶ。同様なファッションで、Yインターフェイス装置24bは、一緒にTNetリンクLyを形成している二つの10ビット・バス30(出力送信に対する)_y及び32_y(入力送信に対する)により(サブプロセッサシステム10Bの)ルータ14Bに接続される。X及びYインターフェイス装置2

4 a, 24 b は、ロックステップで同時の動作され、実質的に同じ時間に同じ動作を実質的に実行する。それゆえに、Xインターフェイス装置24 aだけがデータをバス30_x上に実際に送信するが、同じ出力データがYインターフェイス装置24 bによって生成され、そして誤り検査に用いられる。Yインターフェイス装置24 b出力データは、それがXインターフェイス装置24 aによって受信されかつXインターフェイス装置によって生成される同じ出力データに対して比較されるところのクロス・リンク34_yによってXインターフェイス装置24 aに結合される。このように、CPU12 aのXポートで利用可能にされる出力データは、誤りについて検査される。

【0038】同じファクションで、CPU12 Aのポートから送信された出力データが検査される。Yインターフェイス装置24 bからの出力データは、10ビット・バス30_yによりYポートに、かつXインターフェイス装置によって生成されたもので検査される9ビット・クロス・リンク34_yによりXインターフェイス装置24 aに結合される。

【0039】上述したように、二つのインターフェイス装置24 a, 24 bは、互いに同期の、ロックステップで動作し、それぞれが同時に同じ動作を実質的に実行する。この理由で、CPU12 AのX及び/又はYポートで受信したデータは、このロックステップ・モードに二つのインターフェイス装置を維持すべく両方のインターフェイス装置24 a, 24 bによって受信されなければならない。それゆえに、一つのインターフェイス装置(24 aまたは24 b)によって受信されたデータは、破線及び(Xインターフェイス装置24 aによってXポートで受信される入力データをYインターフェイス装置24 bに伝達する)9ビット・クロス・リンク接続36_x及び(Yインターフェイス装置24 bによってYポートで受信したデータをXインターフェイス装置24 aに伝達する)36_yに示されるように、他のインターフェイス装置(24 bまたは24 a)に渡される。ある一定のより強固なオペレーティング・システムは、マイクロプロセッサ・システムのコンテキストにおけるフォルトトレラント機能で構成される。この型のマイクロプロセッサ・システムは、ハードウェアまたはソフトウェアによって検出された故障から回復するためにソフトウェアをイネーブルすることによってフォルトトレラント環境を供給する。例えば、米国特許第4,817,091号公報は、各プロセッサが、継続動作の表示をそれにより供給すべく、ソフトウェア制御の下で、システムのプロセッサのそれぞれ(それ自身を含む)に周期的にメッセージを送るようなマルチプロセッサ・システムを教示する。プロセッサのそれぞれは、その通常タスクを実行することに加えて、別のプロセッサに対してバックアップ・プロセッサとして動作する。バックアップ・プロセッサの

一つが同胞(シブリング)プロセッサからのメッセージ表示を受信することに失敗した場合には、それ自身のタスクを実行することに加えて、その同胞(いま動作不能であると考えられる)の動作を取って代わる。あまり強固でないソフトウェアまたはオペレーティング・システムを用いている(即ち、検出された故障から回復するための本質的機能がない)、他のフォルトトレラント技術は、ハードウェア及び検出された誤りから回復すべく動作する論理回路で設計される。

【0040】本発明は、両方の型のソフトウェアに対するハードウェア・プラットフォームを供給することに指向される。それゆえに、強固なオペレーティング・システムが利用可能であるときには、処理システム10は、CPU12 A及び12 Bのそれぞれが独立したファクションで動作するような“シンプレックス”モードで動作すべく構成することができる。CPU s 12は、CPU内部データ経路の種々の重要な地点に誤り検査回路素子を有して構成される。ルータ14は、システム10で相互接続されうる種々のCPU s 12間でプロセッサ間通信を供給すると共に、システムのCPUからI/Oパケット・インターフェイス16によって制御される装置への通信ルートを供給する。誤りが検出されたときには、その誤りから回復する役割は、ほとんどの場合、ソフトウェアに残される。代替的に、あまり強固でないオペレーティング・システム及びソフトウェアに対して、処理システム10は、一対のCPU s (例えば、CPU s 12 A, 12 B)が同期された、ロックステップ・ファクションで動作すべく図1に示すように一緒に結合されるように“デュプレックス”モードで動作すべく構成されているハードウェア・ベース・フォルトトレランスを供給し、実質的に同じ瞬間に同じ命令を実行する。それゆえに、各CPUは、他のチェックとして動作する。CPU s 12の一つが故障を生じる場合には、それは、“フェイラーファースト”でかつ誤りがシステムの残りに拡散しかつそれらを汚染することが許容される前にシャット・ダウンする。他のCPU 12は、二つのタスクを実行すべく動作を継続する。次に、デュプレックス・モード動作は、システム・ハードウェアに故障の効果をマスクさせる。

【0041】データ及び指令記号は、9ビット・データ及び指令記号を含んでいるメッセージ・パケットによって種々のCPU s 12とI/Oパケット・インターフェイス16との間で伝達される。CPU 12の設計を簡略化するために、プロセッサ20は、外側のエンティティ(例えば、別のCPU 12またはI/Oパケット・インターフェイス16を介するI/O装置)と直接通信することから排除される。それよりも、見て分かるように、プロセッサは、メモリにデータ構造を構成しかつインターフェイス装置24に制御を転換する。各インターフェイス装置24は、メモリからのデータ構造をアクセスし

かつメッセージ・パケットに含まれた情報による宛先への通信に対して適切なXまたはYポートを介してそれらを伝送するための直接メモリ・アクセス (DMA) 機能のフォームを供給すべく構成されたブロック転送エンジン (BTE; 図9) を含む。処理システム10の設計は、CPUのメモリ28に外側のソース (例えば、CPU12またはI/O装置) によって読み取られるかまたは書き込まれることを許容する。この理由で、CPU12のメモリ28の外部使用が許可されることを確実にすべく注意が払われなければならない。それゆえに、メモリ28へのアクセスは、アクセス要求が来た場所、要求されたアクセスの型、要求されたアクセスの位置、等のようなファクターを試験することによりアクセスを許容するかまたは排除するアクセス妥当性検査 (確認) 機構によって保護される。アクセス妥当性検査は、以下の図16～図20の説明中に記載されるアクセス妥当性検査表 (AVT) 論理回路によって実施される。

【0042】本発明の種々の態様が、ルータ14を介してI/Oパケット・インターフェイス16とCPU12との間で伝送されるデータ及び指令パケットの構成を用いる。従って、処理システム10の構成の説明を続ける前に、TNetリンクL上で送信されかつルータ14によって送られる (ルート付けられる) データ及び指令記号及びパケットの構成をまず理解することは、有利である。

【0043】— パケット構成：4つの基本メッセージ・パケット型がCPU12とシステムの周辺装置17との間に指令記号及びデータを伝送するために用いられる。図5(a)～図5(d)は、そのパケットのフィールドの分解と共に、一つのメッセージ・パケットの構成を示す；図6～図8は、他の3つのパケット型の構成を示す。TNetエリア・ネットワーク上に書き込みデータを伝送するために用いられるメッセージ・パケット型は、HADCパケットとして識別され、かつ図5(a)に示されている。図示するように、HADCパケットは、4つのフィールドを有する：8バイト・ヘッダ・フィールド、4バイト・データ・アドレス・フィールド、Nバイト・データ・フィールド (より大きなデータは、単一パケットによって移動することができるということが明らかであるが、Nは、64の最大値であるのが好ましい)、及び4バイト・サイクリック冗長チェック (CRC) フィールド。図5(b)に詳細に示される、ヘッダ・フィールドは、3バイト宛先IDを含み、メッセージ・パケットの最終宛先を識別する；メッセージ・パケットのソースまたは送信者、トランザクションの型

(例えば、読取りまたは書き込み動作) 及びメッセージ・パケットの型 (例えば、データに対する要求か否か、またはデータ要求への応答か否か) を識別する3バイト・ソースID。宛先IDは、4つのサブフィールドを含む：宛先が配置される“領域”を特定するための領域

IDを含む14ビット・サブフィールド；識別された領域内の宛先を特定している、装置IDを含んでいる6ビット・サブフィールド；及び二つの経路間を選択するために用いられる経路選択 (P) ビット；及び更なる拡張のために確保された3ビット。同様に、ソースIDは、3つのサブフィールドを有する；送信者の領域を識別する、14ビット領域ID；その領域内の送信装置を識別する、6ビット装置ID；及びトランザクションの型を識別する、上記したような、4ビット型サブフィールド。更に、制御フィールドは、9ビット指令/データ“記号”によるメッセージ・パケットの添付したデータ・フィールドに含まれるデータの量を特定する。(各記号は、以下に示すように、指令記号としてデータバイトを、またはその逆を、表わすことができる、単一ビット誤りに対して保護すべく9ビット量としてコード化されたデータの8ビット・バイト誤りである。)

【0044】図5(b)に特に示さないのは、宛先フィールドが3つのサブフィールド、即ち、領域 (Region)

(14ビット)、装置 (Device) (6ビット) 及び経路選択 (Path Select) (1ビット) に分割されるという事実である。領域フィールドの情報は、メッセージ・パケットの宛先を含んでいる特定の“領域”を識別する。領域の概念は、ルータ14のアーキテクチャの説明に関連して以下に説明する。6ビット装置フィールドは、メッセージ・パケットの最終宛先である特定の装置 (例えば、装置17、CPU12、またはMP18) を識別する。領域及び装置フィールドは、宛先を漸増的かつ一意に識別する。経路選択ビットとして確保されたビットは、メッセージ・パケットの宛先を含んでいる (図1に示したような) 二つの“側”XまたはYの一つまたは他のものを識別すべく動作する。経路選択ビットは、メモリ・アクセス妥当性検査 (図16及び図17) 及びルータのポート選択動作 (図31) に関連して以下に更に説明する。残りの3ビットは、必要により更なる拡張のために確保されている。4バイト・データ・アドレス・フィールドは、図5(c)に詳細に示される。アドレス・フィールドは、HADCパケットの場合に、データの添付したNバイトが書き込まれる宛先の仮想位置を識別する。例えば、メッセージ・パケットのソースが、CPU12のメモリ28に書き込まれるべきデータを含んでいるI/O装置17であるならば、データ・アドレス・フィールドは、データが書き込まれるべきメモリ28における位置を識別しているアドレスを含む。(見て分かるように、CPU12に対してデータ・アドレスは、AVT論理回路

(図16) によってメモリ28をアクセスするために実際に用いられる物理アドレスに変換される。I/Oパケット・インターフェイス16は、同様な妥当性検査及び変換機構を有する。) アドレス・フィールドがCPU12のメモリ位置を識別するときには、フィールドは、二つのサブフィールドを含む：アドレス・フィールドの上

位20ビットは、20ビット・メモリ頁数を形成する；残りの12ビットは、メモリ頁にオブセットを形成する。頁数は、妥当性検査情報を含むエントリを含んでいる表の中へのインデックス（索引）としてAVT論理回路（図16）によって用いられる。

【0045】説明したように、HADCメッセージ・パケットは、処理システム10の端末装置（例えばCPU12）間で書き込みデータを伝達すべく動作する。しかしながら、他のメッセージ・パケットは、それらの機能及び使用により、異なって構成されうる。それゆえに、図6は、ヘッダ、アドレス、及びCRCフィールドだけを含んでいるHACメッセージ・パケットを示す。HACパケットは、システム・コンポーネント（例えば、I/O装置17）に読取りデータ要求を送信するために用いられる。図7は、8バイト・ヘッダ・フィールド、Nバイト・データ・フィールド（再度、いかなる整数でもありうるが、Nは、64までである）、及び4バイト・CRCフィールドを有している、HDC型のメッセージ・パケットを示す。HDCメッセージ・パケットは、要求されたデータのリターン（戻り）を含む、読取り要求への応答を伝達するためである。図8は、8バイト・ヘッダ、及び4バイトCRCだけを含んでいる、HCメッセージ・パケットを示す。HCメッセージ・パケットは、データを書込むための要求に肯定応答するために用いられる。

【0046】— インターフェイス装置：X及びYインターフェイス装置24（即ち、24a及び24b—図4）は、CPU12内の3つの主な機能を実行すべく動作する：プロセッサ20をメモリ28にインターフェイスすること；プロセッサに対して透過的に動作するが、その制御下で、I/Oサービスを供給すること；及び外側ソースからのメモリ28へのアクセスに対する要求の妥当性を検査すること。まずインターフェイス機能に関して、X及びYインターフェイス装置24a、24bは、読取られた／書込まれたデータのフェイルーフアースト検査を含むような方法でデータを書込みかつ読取るためにプロセッサ20a、20bをメモリ・コントローラ（Mc s 26a、26b）及びメモリ28にそれぞれ伝達すべく動作する。例えば、書込み動作は、後で検索（読取り）されたときに、適当なデータが検索されるだけでなく、適切なアドレスから検索されたことが分かるように、メモリ28に書込まれたデータだけでなく、そのデータが書込まれる位置のメモリ・アドレスも、見て分かるように、カバーする誤り訂正コード（ECC）を発生すべくそのインテグリティ（及び同時に、インターフェイス装置24が動作すること）を確実にすべく書込まれるべきデータをクロス・チェックするために協同している二つのインターフェイス装置24a、24bを有する。

【0047】I/Oアクセスに関して、プロセッサ20

は、入力／出力システムと直接伝達するための機能を備えていない；それよりも、それらは、メモリ28にデータ構造を書込み、次にそれらのデータ構造を検索すべく直接メモリ・アクセス（DMA）動作を実行するインターフェイス装置24へ制御を渡し、そして所望の宛先への伝達のためにTNetにそれらを渡さなければならない。（宛先のアドレスは、データ構造それ自体に示される。）

X及びYインターフェイス装置24の第3の機能である、メモリ28へのアクセス妥当性検査は、インターフェイス装置によって保守されるアドレス妥当性検査及び変換（AVT）表を用いる。AVT表は、アクセスが許可された各システム・コンポーネント（例えば、I/O装置17、またはCPU12）に対するアドレス、許可されたアドレスの型、及びアクセスが許可されるメモリの物理位置を含む。表は、入力メッセージ・パケットに含まれたアドレスが仮想アドレスなので、アドレス変換を実行するための手段にもなる。これらの仮想アドレスは、メモリ28をアクセスするためにメモリ制御装置26により認識可能な物理アドレスにインターフェイス装置によって変換される。

【0048】図9を参照すると、CPU12AのXインターフェイス装置24aの簡略化されたブロック図が示されている。コンパニオンYインターフェイス装置24b（並びにCPU12B、または他のCPU12のインターフェイス装置24）は、実質的に同じように構成される。従って、インターフェイス装置24aの説明が処理システム10の他のインターフェイス装置24にも同様に適用されるということが理解されるであろう。図9に示すように、Xインターフェイス装置24aは、プロセッサ・インターフェイス60、メモリ・インターフェイス70、割込み論理回路86、ブロック転送エンジン（BTE）88、アクセス妥当性検査及び変換論理回路90、パケット送信機94、及びパケット受信機96を含む。

【0049】— プロセッサ・インターフェイス：プロセッサ・インターフェイス60は、プロセッサ20aとXインターフェイス装置24aとの間の情報フロー（データ及び指令）を処理する。64ビット・アドレス及びデータ・バス（SysAD）23aと9ビット指令バス23bとを含んでいる、プロセッサ・バス23は、プロセッサ20a及びプロセッサ・インターフェイス60を互いに結合する。SysADバス23aは、通常の時分割ファッションで、メモリ・アドレス及びデータを運ぶと同時に、指令バス23bは、指令及びデータ識別子情報（SysCmd）を運び、SysADバス23a上で実質的に同時に運ばれた指令を識別しかつ認定する。プロセッサ・インターフェイス60は、プロセッサ・インターフェイスのメモリまたは制御レジスタに読取り／書き込みを渡すためにプロセッサ装置20aによって発行さ

れた指令を解釈すべく動作する。更に、プロセッサ・インターフェイス60は、(メモリ・コントローラ26を介して)メモリ28へのアクセスに対してアドレス及びデータをバッファする一時記憶装置(図示省略)を含む。メモリから読取ったデータ及び指令情報は、プロセッサ装置20aへの途中で同様にバッファされ、かつプロセッサ装置がそれを受容する準備ができたときに利用可能になる。更に、プロセッサ・インターフェイス60は、Xインターフェイス装置24aに対して必要割込み信号を生成すべく動作する。

【0050】プロセッサ・インターフェイス60は、双方向性64ビット・プロセッサ・アドレス/データ・バス76によってメモリ・インターフェイス70及び構成レジスタ74に接続される。構成レジスタ74は、Xインターフェイス装置24aの他のコンポーネントに含まれる種々の制御レジスタの記号的表現であり、かつそれらの特定コンポーネントを説明するときに説明する。しかしながら、図9に特に示されていないが、構成レジスタ74の種々のものがXインターフェイス24aを実施するために用いられる論理回路の他のものの中に分散するという事実により、プロセッサ・アドレス/データ・バス76は、同様にそれらのレジスタに対する読取りまたは書込みに結合される。構成レジスタ74は、プロセッサ20aに対して読取り/書込みアクセス可能である；それらは、Xインターフェイス装置を“個人専用化”させる。例えば、一つのレジスタは、CPU12Aから生ずるメッセージ・パケットのソース・アドレスを形成するために用いられる、CPU12Aのノード・アドレスを識別する；別の、読取り可能専用は、インターフェイス装置24の固定識別番号を含み、かつ他のレジスタは、例えば、(データ構造及びBTE指令/制御語が配置される)BTE88、(メッセージ・パケットを介して受信した外部で生成された割込みについての情報を含む割込みキューを指している)割込み論理回路86、またはAVT論理回路90によって用いることができるメモリの領域を画定する。他のレジスタは、割込み論理回路86による割込みポスティング(記入)に対して用いられる。レジスタの多くは、それらを採り入れている論理回路コンポーネント(例えば、割込み論理回路86、AVT論理回路90、等)を説明するときに以下に更に説明する。

【0051】メモリ・インターフェイス70は、二つの36双方向ビット・バス25a、25bを含むバス25によってXインターフェイス装置24aをメモリ・コントローラ26(及びYインターフェイス装置24b；図4参照)に結合する。メモリ・インターフェイスは、プロセッサ装置20、BTE88、及びAVT論理回路90からのメモリ・アクセスに対する要求間を調停(arbitrate)すべく動作する。プロセッサ装置20aからのメモリ・アクセスに加えて、メモリ28は、例えば、I/O

装置17からプロセッサ装置20aによって読取られるべく要求されたデータを記憶するために、処理システム10のコンポーネントによってもアクセスされうるか、またはメモリ28は、プロセッサ装置によってメモリに予めセットアップされたI/Oデータ構造に対してもアクセスされうる。これらのアクセスは、全て非同期なので、それらは、調停されなければならない、かつメモリ・インターフェイス70は、この調停を実行する。メモリ28からアクセスされたデータ及び指令情報は、メモリ読取りバス82によってメモリ・インターフェイスからプロセッサ・インターフェイス60に、並びに割込み論理回路86、ブロック転送エンジン(BTE)88、及びアクセス妥当性検査及び変換(AVT)論理回路90に結合される。以下に詳述するように、データは、倍長語量でメモリ28に書込まれる。しかしながら、X及びYインターフェイス装置24a及び24bの両方のメモリ・インターフェイス70は、(64ビット)倍長語を形成しかつバス25に適用し、各メモリ・インターフェイス70は、その64ビット倍長語量の32ビットだけを書込む役割を果たす；メモリ・インターフェイス70によって書込まれない32ビットは、それらが誤りに対して同じ32ビットと比較されるコンパニオン・インターフェイス装置24によってメモリ・インターフェイスに結合される。

【0052】ちょっと脇道にそれると、図1～図3の処理システムでは、割込みは、特定割込み型を伝達するために専用信号回線の従来技術を用いるよりも、メッセージ・パケットとして送信される。割込み情報を含んでいるメッセージ・パケットが受信されるときには、その情報は、CPU12Aの内部で生成された割込みと共に、プロセッサ20による作用に対して処理しかつポスティングするために割込み論理回路86に運ばれる。内部で生成された割込みは、割込みの原因を示している、(割込み論理回路86の内部の)レジスタ71にビットをセットする。次に、プロセッサ20は、割込みにより読取りかつ作用することができる。割込み論理回路は、以下により完全に説明する。Xインターフェイス装置24aのBTE88は、直接メモリ・アクセスを実行すべく動作し、かつプロセッサ20に外部資源をアクセスさせる機構を供給する。BTE88は、プロセッサ20に透過である、I/O要求を生成すべくプロセッサ20によってセットアップし、かつ要求が終了したときにプロセッサに知らせることができる。BTE論理回路88は、以下に更に説明する。入力メッセージ・パケットに含まれるメモリ・アクセスに対する要求は、AVT論理回路90によって妥当性が検査される。アクセス要求の妥当性検査は、要求のソースの識別、要求されたアクセスの型を含んでいる、種々の許可により行われる。更に、AVT論理回路は、要求が適切に確認されたときに実際のアクセスにするために用いることができる物理メモリ・ア

ドレスに、メッセージ・パケットに含まれたときには仮想アドレスである、アクセスが所望である（受信したメッセージ・パケットに含まれる）メモリ・アドレスを変換する。また、AVT論理回路90は、以下により詳細に説明する。

【0053】BTE論理回路88は、送られるべきデータ及び／又は指令記号をパケット送信機94に供給すべくAVT論理回路90に関連して動作する。次に、パケット送信機94は、メッセージ・パケット・フォームにBTE及びAVT論理回路88、90から受信した情報をアSEMBルし、それらが送信できるまでそれらをバッファする。更にまた、BTE及びAVT論理回路88、90は、入力メッセージ・パケットを受信し、解釈しかつ処理すべくパケット受信機96で動作し、必要によりそれらをバッファし、かつメモリ28に記憶するために必要な8バイト幅フォーマットにそれらを変換する。プロセッサから発生されたトランザクション要求を含んでいる出力メッセージ・パケット（例えば、I/O装置からブロック・データを要求している読取り要求）は、要求トランザクション論理回路（RTL）100によって監視される。RTL100は、要求が所定の期間内に応答するかどうかを確認するアウトバウンド（出発）要求に対するタイムアウト・カウンタを供給する；もしそうでなければ、RTLは、要求が受け入れられなかったことをプロセッサ20に報告すべく（割込み論理回路86によって処理されかつ報告される）割込みを生成する。加えて、RTL100は、応答を確認する。RTL100は、応答に対するアドレスを保持し、かつそれが応答を位置決めできるようにプロセッサ20に既知な位置で応答をメモリ28に配置することができるように応答が受信されたときにこのアドレスをBTE88に進める。

【0054】CPU12のそれぞれは、説明するように、多数の方法で検査される。一つのそのような検査は、各CPUのインターフェイス装置24a、24bの動作の前進(on-going)監視である。インターフェイス装置24a、24bは、ロックステップで動作するので、同期性検査は、それらの内部状態のあるものの連続比較によりペアになったインターフェイス装置24a、24bの動作状態を監視することによって実行できる。このアプローチは、CPU12Aの装置24aに含まれる状態マシン（図示省略）の1ステージを使用し、かつそのステージによって想定された各状態をインターフェイス装置24bのその同じ状態マシン・ステージと比較することによって実施される。インターフェイス装置24の全ての装置は、それらの動作を制御すべく状態マシンを用いる。従って、インターフェイス装置24とMC26との間のデータ転送を制御するメモリ・インターフェイス70の状態マシンを用いるのが好ましい。それゆえに、インターフェイス装置24aのメモリ・インター

フェイス70に用いられる状態マシンの選択されたステージが選択される。また、一つのインターフェイス装置24bの状態マシンの同じステージも選択される。二つの選択されたステージは、インターフェイス装置24a、24b間で伝達されかつインターフェイス24a、24bの両方に含まれる比較回路によって受信される。インターフェイス装置が互いにロックステップで動作すると、状態マシンは、同じ同一状態を通して同様にマーチし、実質的に同じ瞬間で各状態を想定する。インターフェイス装置が誤りに遭遇するか、または故障したならば、そのアクティビティは、インターフェイス装置が発散する原因となり、かつ状態マシンは、異なる状態を想定する。また、状態マシンから比較回路に伝達された選択ステージが異なる場合もある。この相違は、比較回路が、そのCPUのインターフェイス装置24a、24bがもはやロックステップでないCPU12A（または12B）の注意を喚起する“損失sync(lost sync)”誤り信号を発行し、かつそれに応じて作用する原因となる。この技術の一例は、Humphrey, et. al. により発明されかつ本願発明の出願人にアサインされた米国特許第4,672,609号公報に見ることができる。

【0055】図9に戻ると、CPU12AのXインターフェイスのパケット受信機96は、Xポートだけをサービスすべく機能し、サブプロセッサシステム10A（図1）のルータ14Aによって送信されたそれらのメッセージ・パケットだけを受信する。Yポートは、コンパニオン・サブプロセッサシステム10Bのルータ14Bからメッセージ・パケットを受信すべくYインターフェイス装置24bによってサービスされる。しかしながら、両方のインターフェイス（並びにMc26及びプロセッサ20）は、示されたように、基本的に構造及び機能において両方が実質的に同一である互いのミラー・イメージである。この理由で、一つのインターフェイス装置（例えば、24a）によって受信された、メッセージ・パケット情報は、コンパニオン・インターフェイス装置（例えば、24b）へも処理するために渡されなければならない。更に、インターフェイス装置24a、24bの両方は、XまたはYポートからの送信に対して同じメッセージ・パケットをアSEMBルするので、関連ポート（例えば、Yポート）から実際に伝達されるインターフェイス装置（例えば、24b）によって送信されるメッセージ・パケットは、誤りに対するクロスチェックのために他のインターフェイス装置（例えば、24a）にも結合される。これらの特徴は、図10及び13に示されている。

【0056】— パケット受信機：図10を参照すると、X及びYインターフェイス装置24a、24bのパケット受信機96（96x、96y）の受信部分は、拡大して示されている。示したように、各パケット受信機96x、96yは、TNetリンク3.2の対応している

ものを受信すべく結合されたクロック sync (CS) FIFO102を有する。CS FIFOs102は、パケット受信機96の局所クロックに入力指令/データ記号を同期すべく動作し、それらをバッファして、それをマルチプレクサ(MUX)104に渡す。しかしながら、Xポート及びXインターフェイス24aのパケット受信機96xで受信した情報は、MUX104xに渡されることに加えて、クロスリンク接続36_xによりYインターフェイス装置24bのパケット受信機96yのMUX104yに結合されるということに注目する。同様なファッションで、Yポートで受信した情報は、クロスリンク接続36_yによりXインターフェイス装置24aに結合される。この方法で、対応Y、Yインターフェイス装置によりX、Yポートの一つで受信された情報パケットの指令/データ記号は、両方が同じ情報を処理しかつインターフェイス装置24の他のコンポーネント及び/又はメモリ28に伝達するように他のものに渡される。

【0057】図10を続けると、どのポートX、Yがメッセージ・パケットを受信しているいかにより、MUXs104は、インターフェイス装置24の記憶及び処理論理回路110への通信に対してCS FIFOs102x、102yの一つまたは他のものの出力のいずれかを選択する。各9ビット記号に含まれる情報は、その符号化が図14を参照して以下に説明される、指令またはデータ情報の8ビット・バイトである。記憶及び処理論理回路110は、まず9ビット記号を8ビットデータまたは指令バイトに変換し、64ビット倍長語としてバイトを編成し、よのように形成された倍長語を入力パケット・バッファ(特に示していない)に渡す。入力パケット・バッファは、それをメモリ・インターフェイス70、並びにAVT論理回路90及び/又はBTE88に渡すことができるまで受信した情報を一時的に保持する。パケット受信機96は、それぞれメッセージ・パケットのCRCを検査するCRCチェッカ論理回路106を含む。各CRCチェッカ論理回路106は、どのポート(XまたはY)がメッセージ・パケットを受信するかに関係なく、両方の受信機96x、96yが受信したメッセージ・パケットのCRCを検査するように配置されるということに、特に、注目する。この特徴は、フォルト分離機能を有する。この受信ステージで検査されるけれども、他のものからではなく一つの受信機からのCRC誤り表示は、二つの受信機間のインターフェイスまたは誤りを発行している受信機の論理回路における問題を示す。それゆえに、フォルトは、受信しているCS FIFOの出力からの経路のその部分に対して少なくとも最初のうちは分離することができる。

【0058】図示されていないのは、CS FIFOs102x、102yの出力は、MUX104に加えて指令復号装置にも結合されるという事実である。指令復号

装置は、指令記号を認識すべく(以下に説明する方法でデータ記号からそれらを区分すべく)動作し、受信機制御装置、パケット受信機動作を制御すべく機能する状態マシン・ベース素子に印加される指令信号をそれから生成すべく指令記号を復号する。上述したように、パケットは、周期冗長検査(CRC)値によって誤り保護される。それゆえに、受信したパケットのCRC情報がMUX104の出力に現れるときには、記憶制御装置の受信機制御部分は、CRC記号を計算するためにCRC検査論理回路106をイネーブルし、同時に、データ記号は、受信したCRCに対する生成された量をメッセージ・パケットと続いて比較すべく受信される。可能な誤りがパケット受信機96への送信中に生じたことを示している、ミスマッチが存在するならば、CRC検査論理回路106は、割込みレジスタ(割込みレジスタ280; 図21)をセットするために用いられる誤り割込み信号(BADCRC)を発行しかつパケットは、廃棄される。しかしながら、パケット・ヘッダは、後の試験のために割込みキューにセーブされる。

【0059】以下に更に説明するように、CS FIFOsは、インターフェイス装置24のパケット受信機96においてだけでなく、ルータ14及びI/Oパケット・インターフェイス16の各受信ポートでも見出される。しかしながら、CPU s12A、12Bとルータ14A、14B(即ち、ポート1及び2)を接続するTNetリンクLから記号を受信するために用いられたCS FIFOsは、ルータ14の他のポート上で用いられるもの、及びCPU12に直接接続されていない他のルータ14とは多少異なる。換言すると、周波数ロックド(frequency locked)クロッキングを用いている素子間で記号を伝達するために用いられたCS FIFOsは、近周波数クロッキングを用いている素子間で記号を伝達するために用いられたものとは異なる。また、以下の説明は、CS FIFOsは、近周波数モード(即ち、送信及び受信素子のクロック信号は、同じである必要はないが、所定のトレランス内であることが期待される)で動作している素子間でTNetリンクL上の情報を転送することにおける重要部分の役をするということも明らかにする。しかし、CS FIFOsは、一対のサブプロセッサシステムがデュプレックス・モードで動作しかつサブプロセッサシステム10A、10Bの二つのCPU s12A及び12Bが同期された、ロックステップで動作し、同時に同じ命令を実行するときには、より重要な部分の役をし、かつ固有の機能を実行する。この後モードで動作しているときには、ルータ14Aまたは14Bの一つからCPU s12A及び12Bに送信された情報が、同期、ロックステップ動作を維持するために本質的に同時に両方のCPU sによって受信されることは、絶対必要である。ルータ14A及び14Bのクロッキング・レジムがCPU s12A及び12Bのものに

正確に同期されることを確実にするのは、周波数封じ込みクロッキングを用いているときでさえも、非常に難しいので、これは、不幸にも、容易なタスクではない。CPU 12のパケット受信機96において、CPU 12に記号を送信するために用いられるルータ14のクロックとそれらの記号を受信すべくインターフェイス装置24によって用いられるクロックとの間の可能な相違を収容する(accommodate)ことは、CS FIFO 102の機能である。

【0060】CS FIFO 102の構造は、図に、説明の目的で、図式的しめされている；CS FIFOの好ましい構造は、図12に示されている。再び、CS FIFOに対してここで参照がなされるときには、特に示さない限り、図11を参照して説明される機能及び動作、及び図12に示された構造を有している構造を参照することを意図するということが理解されるべきである。従って、図11のCS FIFOの説明は、その性質において一般的であることを意図しており、そのように理解されるべきである。更に、上記したように、周波数ロック動作に用いられるCS FIFOsのあるものは、近周波数動作で用いられるものと異なるが、以下の説明は、両方に適用する。その説明に続いては、近周波数環境における動作のためにCS FIFOの一般的な構成になされなければならない変更の説明である。図11に示すのは、パケット受信機96xのCS FIFO 102xである。CS FIFO 102yは、CS FIFO 102xの以下の説明がCS FIFO 102yに同様に適用されることが理解されるような実質的に同一の構成及び動作である。図11では、CS FIFO 102xは、ルータ14A(図1)の送信(Xmt)レジスタ120から送信される9ビット指令/データ記号及びルータからの添付送信クロック(T_Clk)を受信すべくTNetリンク32xによって結合されて示される。図11の破線Bは、対応TNetリンク32xの一端における送信エンティティ(ルータ14A)とCPU12Aの受信エンティティ、パケット受信機96xとの間のクロック境界を記号化する。従って、CS FIFO 102xは、記憶キュー126へ渡される前にそれらが一時的に保持される(例えば、一つのT_Clk期間に対して)受信(Rcv)レジスタ124で9ビット記号を受信する。記憶キュー126は、表示及び説明の容易のために4つの位置を含んで示される。しかしながら、追加記憶位置が供給でき、かつ事実必要または所望でありうることは、この分野における当業者には、明らかである。

【0061】受信した記号は、プッシュ・ポインタ・カウンタ128によって識別された記憶キュー126の位置で(Rcvレジスタ124から)CS FIFO 102x上に“プッシュ”される。プッシュ・ポインタ・カウンタ128は、T_Clkによってクロックされ

る、バイナリ・カウンタ(2進計数器)の形であるのが好ましい。次に、受信した記号は、プル・ポインタ・カウンタ130によって識別される記憶キュー123の位置から逐次的に“プル(pull)”され、FIFO出力レジスタ132に渡される。局所クロック信号、“Rcv Clk”は、記憶キュー126及びFIFO出力レジスタ130から記号をプルするために用いられ、(CPU 12Aに対して)内部で生成された信号によって生成される。FIFO出力レジスタ132からの記号は、MUX104xに行く。TNet送信に用いられたプロトコルにより、記号の一定ストリームは、常に全ての送信ポート(例えば、CPU12aのX及びYポート、ルータ14AまたはI/Oインターフェイス16のいずれかの送信ポート—図1)から送信される；それらは、ある一定の状況(例えば、リセット、初期化、同期化及び以下に説明するその他)の間を除いて—実際の指令/データ記号(例えば、パケット)またはIDLE記号のいずれかでありうる。上述したように、ルータ14Aの送信レジスタ120に保持された各記号は、Rcvレジスタ124に結合され、かつルータ14Aによって供給されるクロック信号、T_Clkで、記憶キュー126に記憶される。逆に言えば、記号は、局所的に生成されたクロック、Rcv Clkと同期して記憶キュー126からプルされる。これらは、実質的に同じ周波数であるにもかかわらず、二つの異なるクロック信号である。しかしながら、CS FIFO 102xに入ってくる記号とCS FIFOからプルされるその同じ記号との間に十分な時間(例えば、2〜3のクロック)が存在する限り、準安定性問題が存在すべきではない。入力クロック信号(T_Clk)及びRcv Clkが周波数封じ込みモードで動作されるときには、CS FIFO 102xは、決してオーバーフローまたはアンダーフローすべきでない。

【0062】CS FIFO 102xを初期化するのは、次の通りである。アウトセットでは、ルータ14Aは、送信クロック信号、T_Clkの各パルスに対するIDLE記号を送信し、IDLE記号でRcvレジスタ124、記憶キュー126、及びFIFO出力レジスタ132を最後に充たし、CS FIFO 102xをアイドル状態にリセットする。プッシュ・ポインタ・カウンタ128及びプル・ポインタ・カウンタ130は、SYNC指令記号の受信(及び検出)によりリセットされる。SYNC信号の受信は、プッシュ・ポインタ・カウンタ128を、記憶キュー126の特定の位置に対するポイントにセットする原因になる。同時に、プル・ポインタ・カウンタ130は、好ましくは2つの記憶位置によってプッシュ・ポインタ・カウンタのそれから離間した記憶キュー126の位置におけるポイントに同様にセットされる。それゆえに、公称(nominal)2クロック遅延は、記憶キュー126に入ってくる記号と記憶キュー

を出て行くその同じ記号との間に確立され、それがクロック・アウトされかつMUX 104x (及び104y) によって記憶及び処理装置110x (及び110y) に渡される前に記憶キュー126に入ってくる各記号を整定(settle)させる。送信及び受信クロックは、位相一独立であるので、公称2クロック遅延は、許容されたリセット・スキューが1クロック以下であるべく期待されるようにある所定量が+または-の誤りを含む。

【0063】図12は、各組合せが記憶キュー126の記憶位置を形成している、マルチプレクサ/ラッチ組合せ140、142によって形成されているとして記憶キュー126を示している、CS FIFO 102xの一つの実施を示す。ラッチ142は、T_{CLK}の各パルスでクロックされる。プッシュ・ポインタ・カウンタ128は、その関連ラッチ142に結合されるべくマルチプレクサ140の一つにrcvレジスタ124の出力を選択させるべくデコータ144によって復号される。ラッチは、T_{CLK}と、Rcvレジスタを関連ラッチ142に伝達させるためにマルチプレクサ140の別のものをもたらしべくインクリメントされたプッシュ・ポインタ・カウンタとで装填される。rcvレジスタ124の出力を受信すべく選択されないそれらのラッチ142は、その代わりにT_{CLK}を有するラッチの内容を受信しかつ装填する。実質的に同じ時間に、プル・カウンタ130は、各Rcv CLKで - FIFO出力レジスタ132に転送されかつFIFO出力レジスタ132によって装填されるべく、マルチプレクサ146を介して、ラッチの一つの内容を選択する；プル・ポインタ・カウンタは、同時に、更新される(インクリメントされる)。

【0064】CS FIFO 102xは、一対のCPUs 12がデュプレックス・モードで機能しているときにだけ、かつルータ14A、14BとペアになったCPUs 12A、12Bとの間の送信に対してだけ用いられる周波数封じ込みクロッキング(即ち、T_{CLK}及びRcv CLKは、周波数において実質的に同じであるが、同相である必要はない)を実施すべく構成される

(図1、図2及び図3)。CPUs 12と通信していない(デュプレックス・モードで機能している)ルータ14(及びI/Oインターフェイス16)の他のポートは、近周波数クロッキングで記号を送信すべく動作する。そうであっても、クロック同期化FIFOsは、近周波数クロッキングで送信された記号を受信すべくこれらの他のポートで用いられ、これらクロック同期化FIFOsの構造は、周波数封じ込み環境で用いられたもの、即ち、CS FIFOs 102のものと同質的に同じである。しかしながら、違いが存在する。例えば、記憶キュー126の記号位置は、9ビット幅である；近周波数環境では、クロック同期化FIFOsは、10ビット幅であるキュー126の記号位置を用い、余分のビ

ットは、その状態に依存して、関連記号が有効か否かを識別する、“有効”フラグである。この特徴は、この議論において更に説明する。

【0065】ルータ14は、記号を送信または受信すべくルータ14のものと同じ公称周波数であるが、多少異なる実周波数を有する、他のクロック・ソースの保護(影響)の下で走る他のキャビネットの装置(例えば、他のルータまたはI/Oインターフェイス16)と通信しているそれ自身をしばしば見出しうる。これは、近周波数状況であり、かつこれは、記号転送に対するクロッキングのこの形は、デュプレックス・モードであるときにCPU 12に直接接続するそれらのポートを除くルータ14の全ポートで見られる。近周波数モードでは、クロック信号(例えば、一端で記号を送信するために用いられるクロック、及び他端で記号を受信するために用いられるクロック)は、他に対してサイクルの結果としてゲインする一つでゆっくりドリフトしうる。これが起きるときには、CS FIFO 102の二つのポインタ(それぞれプッシュ及びプル・ポインタ・カウンタ128、130)は、どのエンティティ(送信機または受信機)がより速いクロック・ソースを有するかによって、より近い記憶キュー126の一つの記号位置または互いにもっと離れた一つの記号位置のいずれかをポイントする。このクロック・ドリフトを処理するために、二つのポインタは、周期的に効果的に再同期される。

【0066】CPUs 12がペアにされかつデュプレックス・モードで動作しているときには、全ての4つのインターフェイス装置24は、同じデータを送信しかつ同じクロック(T_{CLK}及びRcv CLK)でデータを受信すべく、ことのほか、ロックステップで動作し、周波数封じ込みクロッキングが必要でありかつ用いられる。CPUs 12がシンプレックス・モードで動作するときには、それぞれは他とは独立であり、クロッキングは、近周波数であるだけが必要である。インターフェイス装置24は、Rcvレジスタ124を初期化しかつ送信ルータ14に同期するためにSYNC指令記号との組合せで用いられるSYNC CLK信号を受信する。記号転送に対して近周波数または周波数封じ込みクロッキング・モードのいずれかをを用いるときには、CS FIFO 102xは、ある既知の状態から開始するのが好ましい。入力記号は、パケット受信機96の記憶及び処理装置110によって試験される。記憶及び処理装置110は、指令記号を探し、かつそれにより適切に作用する。ここで関連するのは、パケット受信機96がSYNC指令記号を受信するときにそれが記憶及び処理装置110によって復号されかつ検出されることである。記憶及び処理装置110によるSYNC指令記号の検出は、RESET信号のアサーションをもたらす。RESET信号は、SYNC CLK信号の同期制御下で、(クロック同期化バッファを含んでいる)入力パッ

ファを所定の状態にリセットし、かつそれらをルータ14に同期すべく用いられる。

【0067】ルータ14A、14Bの一つまたは両方のインターフェイス装置24のCSFIFOs 102の同期化は、同期化を説明するセクションにおいて以下により完全に説明する。

【0068】— パケット送信機：各インターフェイス装置24は、CPU12のXまたはYポートの一つだけから送信しかつそこで受信すべく割り当てられる。インターフェイス装置24の一つが送信するときには、他は、送信されるデータを検査すべく動作する。シンプレックス・モードで動作するときでさえもCPU12に自己検査フォルト検出及びフォルト包括機能を供給するので、これは、パケット送信機の重要な機能（特徴）である。この機能（特徴）は、X及びYインターフェイス装置24a、24bのパケット送信機94x、94yを、短縮形で、それぞれ表わす、図13に示されている。両方のパケット送信機は、同じように構成され、一つ（パケット送信機94x）の説明は、特に示したことを除き、他（パケット送信機94y）に同様に適用される。図13に示すように、パケット送信機94xは、倍長語

（64ビット）フォーマットで一送信されるべきデータを、関連インターフェイス装置（ここでは、Xインターフェイス装置24a）のBTE88またはAVT90から、受信するパケット・アセンブリ論理回路152を含む。パケット・アセンブリ論理回路152は、CPU12のXまたはYポートからの送信の準備ができるまで情報をバッファし、倍長語フォーマットからバイト・フォーマットにデータを変換すべくバイト・ステアリング動作を実行し、パケット・フォーマットにバイトをアセンブルし、かつXまたはYエンコーダ150x、150yの一つにそれらを渡す。エンコーダ150の一つだけが、どのポート（XまたはY）が合成メッセージ・パケットを送信するかによって、バイトを受信する。

【0069】8ビット・バイトを受信するXまたはYエンコーダ150は、図14に示した9ビット指令／データ記号にそれを符号化すべく動作する。合成9ビット記号の3つの左側ビットの符号化が以下の表1の3つの最左列に示される。

【0070】

【表1】

表 1
8B-9B 記号符号化

CDC	CDB	CDA	機 能
0	0	0	指令
0	0	1	誤り
0	1	0	誤り
1	0	0	誤り
0	1	1	データ<7:6>=00
1	0	1	データ<7:6>=01
1	1	0	データ<7:6>=10
1	1	1	データ<7:6>=11

【0071】表1が示すように、図14に関連して見ると、9ビットの高位3ビット（CDC、CDB、CDA）は、記号の残りの、下位6ビット（CD5、CD4、CD3、CD2、CD1、及びCD0）が（1）指令情報または（2）データとして解釈されるべきかを示すべく符号化される。結果として、3つの上位ビットCDC、CDB、及びCDAが全てゼロであるならば、9ビット記号は、指令記号としてそれにより識別され、かつ残りの6ビットは、指令を形成する。例えば、“000cccccc”と表される指令／データ記号は、“c”ビットが指令である、指令として解釈される。他方、指令／データ記号の、3つの上位ビットCDC、CDB、及びCDAが、データを表わす4つの値のいずれかを取るならば、それらは、残りの6ビットのデータと組合わされるべき2ビットのデータとして解釈され、それから1バイトのデータを得る。残りの6ビットは、データバイトの下位ビットである。従って、“11000

1101”と表される、指令／データ記号は、データ記号として解釈され、かつ“10001101”と表される1バイトのデータに変換される。上位3ビットが001、010、及び100の形を取るならば、それは誤りである。

【0072】指令記号からそのデータ記号を分離する3つの誤りコードは、指令及びデータ間の二つの最小ハミング距離(Hamming distance)を確立する。単一ビット誤りは、データを指令記号にまたはその逆に変えることができない。更に、（データ記号とは反対側の）指令記号の下位6ビットは、指令を含んでいる6ビット位置が正確に3つの“1”を常に含むような既知の“6つのうちの3つ”の符号に符号化される。全ての単一方向誤り、並びに指令記号のあらゆる奇数の誤りは、検出される。データにおける誤りは、指令記号をデータに変える誤りであるように、パケットCRCsを介して検出される。データを指令記号に変える誤りは、以下により完全に説明

されるように、CRC及び／又はプロトコル違反（妨害）誤りによって検出される。XまたはYエンコーダ150のどれがパケット・アセンブリ論理回路152から情報のバイトを受信するのかは、取るべき経路を示している経路ビット（P）を含んでいる、送信されるべき情報に含まれる宛先IDに基づく。例えば、情報の宛先IDは、それがCPU12のXポートを介して送られるということを提案するものと想定する。（パケット送信機94x、94yの両方の）パケット・アセンブリ論理回路152は、その情報をXエンコーダ150xに送る；同時に、それは、IDLE記号をYエンコーダ150yに送る。（記号は、X及びYポートから連続的に送られる；それらは、送信される処理におけるメッセージ・パケットを形成する記号、IDLE記号、または制御機能を実行すべく用いられる他の指令記号のいずれかである。）

X及びYエンコーダ150の出力は、マルチプレクサ154、156を含んでいる、マルチプレキシング構造に印加される。マルチプレクサ154の出力は、Xポートに接続する。（インターフェイス装置24bは、マルチプレクサ154の出力をYポートに接続する。）マルチプレクサ156は、クロスリンク34yを介して、Yポートに接続するマルチプレクサ154の出力も受信するチェッカ論理回路160に接続する。Xポート及びTNetリンク30xに接続するマルチプレクサ154の出力は、（インターフェイス装置24bの）パケット送信機94yのチェッカ論理回路160にクロスリンク34xによっても結合されるということに注目する。

【0073】マルチプレクサの選択（S）入力、構成レジスタ162のX/Yステージからの1ビット出力を受信する。構成レジスタ162は、インターフェイス装置24に形成されるOLAP（図示省略）を介してMP18にアクセス可能であり、かつインターフェイス装置24を、特に、“個人専用”にする情報を伴って書込まれる。ここで、構成レジスタ162のX/Yステージは、Xエンコーダ150x出力をXポートに伝達すべく、Xインターフェイス装置24aのパケット送信機94xを構成する；Yエンコーダ150yの出力は、チェッカ160に同様に結合される。同様なファクションで、

（Yインターフェイス装置24bの）Yパケット送信機94yの構成レジスタ162のX/Yステージは、マルチプレクサ154にYエンコーダ150yの出力を選択させ；かつそれがXポート送信と比較されるパケット送信機160のチェッカ160に結合されるべくXエンコーダ150xの出力を選択させるような状態にセットされる。簡単に言うと、XまたはYポートからのメッセージ・パケット送信の動作は、次の通りである。まず、示されたように、メッセージ・パケット送信が存在しないときには、X及びYエンコーダの両方は、制御機能を実行すべく用いられるIDLE記号または他の記号を送信

する。両方のパケット送信機94の構成レジスタ162のX/Yステージが上記したようにセットされると（即ち、マルチプレクサ154によって出力ポート（X）に伝達されたパケット送信機94xのXエンコーダ150x；マルチプレクサ154によってポート（Y）に伝達されたパケット送信機94yのYエンコーダ150y）、（パケット送信機94xの）Xエンコーダ150xからのIDLE記号は、CPU12AのXポートから送信され、かつ（パケット送信機94yの）Yエンコーダ150yによって生成されたIDLE記号は、Yポートから送信される。同時に、Xポート送信は、パケット送信機94yのチェッカ160にクロスリンク34xによって結合され、かつそのパケット送信機のXエンコーダ150xによって生成されたもので検査される。同じ方法で、Yポートから出力するIDLE記号は、パケット送信機94yから、それらがパケット送信機94xのYエンコーダ150yによって生成されたものに対して検査されるパケット送信機94xのチェッカ160に結合される。

【0074】この説明は、重要な事実を明らかにする：パケット送信機は、それらが正しい動作に対して監視されるためにメッセージ・パケットを送信する必要がない。逆に、メッセージ・パケット・トラフィックが存在しないときでさえも、二つのパケット・インターフェイス94（及び、それにより、それらが関連するインターフェイス装置24）の動作は、連続的に監視される。チェッカの一つがそれに印加されたものの間でミスマッチを検出したならば、ERROR信号がアサートされて、プロセッサ20による適切なアクションに対して内部割込みがポスト（表示）されることを結果として生ずる。メッセージ・パケット・トラフィックは、同じような方法で動作する。パケット送信機94のパケット・アセンブリ論理回路152が送信に対して情報を受信し、かつ宛先IDがXポートが用いられるべきであることを示すということを、ここで、想定する。パケット・アセンブリ論理回路は、各バイトを符号化された9ビット形に変換する、両方のインターフェイス装置96のXエンコーダ150xに、一度に1バイトずつ、その情報を進める。パケット送信機94xのXエンコーダ150xの出力は、XポートかつTNetリンク30x、及びパケット送信機94yのチェッカ160にマルチプレクサ154によって伝達される。また、パケット送信機94yのXエンコーダの出力は、それが、パケット送信機94xからのもので検査される、チェッカ160にだけ、マルチプレクサ156によって、結合される。再び、インターフェイス装置24a、24bの動作、及びそれらが含むパケット送信機は、誤りに対して検査される。

【0075】同じファクションで、Yポート・メッセージ・パケット送信が監視されるということをここで理解することができる。図9にちょっと戻り、出力メッセー

ジ・パケットがプロセッサ始動型トランザクション（例えば、読取り要求）であるならば、プロセッサ20は、応答でメッセージ・パケットが戻されることを期待する。それゆえに、BTE88が送られるべきデータをメモリ28からパケット送信機94に転送するときには、それは、要求トランザクション論理回路100の要求タイマ（図示省略）をセットし、その内で応答を受信すべきタイムアウト期間をマーキングすることを要求タイマに開始させる。出力要求に対する応答が受信されたときには、パケット受信機96の応答整合(reply match)回路は、メッセージ・パケットが応答であり、かつ要求タイマがリセットされることを決定する。宛先への顕著な要求の各数に対して一つの要求タイマ（図示省略）だけが存在する。BTE88がトランザクションの送信を開始するたびに、タイマがリセットされる。他方、割り当てられた時間内に応答が受信されなかったならば、要求タイマは、特定のトランザクション（例えば、読取り要求）に対する応答の欠如をプロセッサ20にそれにより報告すべく割込み論理回路（図21）にタイムアウト（時間切れ）信号を発行する。複数の顕著な要求が管理されることが望ましいならば、要求タイマの追加するもの—各顕著な要求に対して一つ—を用いることができる。

【0076】CPU12Aのメモリ28への外部アクセスが供給されるけれども、それは、保護なしではない。メモリ28へのアクセスに対する外部的に生成された要求は、認められ、かつ2～3の例として、要求のソースの識別、要求されたアクセスの型（例えば、読取りまたは書込み）、アクセスのメモリ領域、を含んでいる、ある一定の基準に従って許可されたときにのみ許される。また、アクセスされることが望ましいメモリ装置28のメモリの領域は、仮想またはI/Oメモリ・アドレスによりメッセージ・パケットで識別される（それにより、仮想記憶方法を使用させる）。許可、及び許されるならば、アクセスの決定は、これらの仮想アドレスがメモリ28の物理アドレスに変換されることを必要とする。そして、CPU12Aの外部の装置または素子によって生成された割込みは、プロセッサ20を中断すべくメッセージ・パケットを介して送信され、受信したときにメモリ28にも書き込まれる。これら全ては、割込み論理回路及びAVT論理回路86、90によって処理される。AVT論理回路装置90は、メモリ28へのアクセスが許可された各可能な外部ソースに対するAVTエントリを含んでいる（メモリ28のプロセッサ20によって保守される）表を用いている。各AVTエントリは、指定のソース素子または装置、及びアクセスが認められるメモリの、特定の頁（頁は、標準的に4K（4096）バイトである）、または頁の一部分を識別する。一頁以上がCPU12の外部の素子によってアクセスされるならば、素子によってアクセスされることが望ましい各頁に

対してAVTエントリが存在しなければならない。更に、各AVTエントリは、許可されたメモリ動作の型（例えば、書込み、読取り、または両方）の情報を含む。AVT表は、必要でなく、かつ“期待された”メモリ・アクセスに対して用いられない。期待されたメモリ・アクセスは、I/O装置からの情報に対する読取り要求のような、CPU12（即ち、プロセッサ20）によって始動されたものである。これら後者のメモリ・アクセスは、各プロセッサ始動型要求に割り当てられたトランザクション・シーケンス番号(TSN)によって処理される。読取り要求が生成された頃に、プロセッサ20は、読取り要求に応じて受信されることが期待されるデータに対してメモリの領域を割り当てる。この領域に対するアドレスは、読取り要求が送られるときに要求トランザクション論理回路100によって保守されるレジスタ・ファイル（図示省略）に記憶され、かつアドレスに対するレジスタ・ファイルへのポインタは、TSNである。それゆえに、読取り要求への応答は、データを伴ってリターンし、かつ戻されたデータを記憶すべくメモリのバッファ領域のアドレスを得るためにそれが運ぶTSNを用いる。

【0077】アクセス妥当性検査は、以下のセクションでより完全に説明される。メモリ・アレー28は、事実上、それぞれがメモリ28に書込まれるかまたはメモリ28から読取られる各64ビットの倍長語の半分を管理するメモリ・コントローラ26a、26bによってそれぞれが管理される二つの半分に分割される。次に、メモリ・コントローラ26a及び26bは、各インターフェイス装置24a、24bのメモリ・インターフェイス70にそれぞれ結合される。64ビット倍長語は、“上位”MC26aによって書込まれる上位32ビット（及び関連ECC）及び“下位”MC26bによって書込まれる下位32ビット（及び関連ECC）でメモリ28に書込まれる。Mc s 26a、26bは、各々、インターフェイス装置24a、24b（図9）のそれぞれのメモリ・インターフェイス70（70a、70b）からデータの32ビット及び4ECCチェック・ビットをそれぞれ受信する。図15を参照すると、各メモリ・インターフェイス70は、メモリに書込まれるべき64ビットのデータを、関連インターフェイス装置24の、プロセッサ・インターフェイス60からのバス82またはAVT論理回路90（図9参照）からのバス83のいずれかから受信する。バス76及び83は、どれがMCADバス25に結合されるべきであるかを選択するマルチプレクサ(MUX)34に印加される。

【0078】各メモリ・インターフェイス70a、70bは、同じで、かつ全ての、メモリに書込まれるべき64ビットを受信するが、それぞれは、それら64ビットのデータの半分（及びそれぞれが生成する8ビットのECCチェック・ビットの4つ）だけをMc s 26a、2

6 bに転送する。Mc s 2 6を駆動するために用いられない3 2ビット（及びE C C論理回路8 5によって生成された8ビットのE C Cチェック・ビットの4つ）は、それらの間のクロス検査のために各メモリ・インターフェイス7 0から他に結合される。それゆに、例えば、（インターフェイス装置2 4 aの）メモリ・インターフェイス7 0 aは、6 4ビットのデータの“上位”3 2ビット（及び8ビットE C C検査語の4ビット）だけでM C 2 6 aを駆動する。同時に、メモリ・インターフェイス7 0 aは、そのコンパニオン・メモリ・インターフェイス7 0 bからデータの“下位”3 2ビットを受信し、かつ比較論理回路8 1によりそれ自身の下位3 2ビットとそれを比較する。ミス比較が検出されたならばE R R O R信号がアサートされる。同様なファッションで、コンパニオン・メモリ・インターフェイス7 0 bは、メモリ2 8に書込まれるべく6 4ビット倍長語を伴って供給されるが、下位3 2ビット（及び生成されたE C C検査ビットの4ビット）だけが用いられる。メモリ・インターフェイスは、メモリ・インターフェイス7 0 aから上位3 2ビットを受信し、かつ比較論理回路8 1でそれらをそれ自身の上位3 2ビットと比較して、ミス比較が結果として生じたならばE R R O R信号を発行する。

【0 0 7 9】追加の誤り検査は、各メモリ・インターフェイス7 0のE C C検査回路8 5によって読取り動作上で実行される。M C 2 6から戻された各6 4ビット倍長語は、8つのE C C検査ビットと一緒に、両方のメモリ・インターフェイス7 0によって受信される。データ及びE C C検査ビットは、各メモリ・インターフェイス7 0のE C C論理回路8 5に印加され、通常ファッションでデータのインテグリティを検査するためのシンδροームを生ずる。単一ビット誤りが検出されたならば、E C C論理回路8 5は、訂正（修正）を実行する；訂正不可能な誤りが検出されたならば、E C C論理回路は、割込みレジスタ2 8 0（図2 6）の状態をセットすることを結果として生ずる、誤り信号（図示省略）を発行し、かつ動作の凍結をもたらす。各メモリ・インターフェイスのE C C論理回路8 5によって実施される特定のE C C検査は、1 1 2ビット・フィールドまでのS E C - D E C - S b E Dアクロス(across)に対して8検査ビットを用いる。コードは、奇数列重みコード(odd column weight code)であり、あらゆる単一誤りが奇数のシンδροーム・ビットを生成することを意味する。1 1 2の可能ビットのうち、6 4は、データ、8は、検査ビットであり、4 0ビットを未使用のまま残す。

【0 0 8 0】 - アクセス妥当性検査：先に示したように、C P U 1 2 Aの外部の処理システム1 0のコンポーネント（例えば、I/Oパケット・インターフェイス1 6の装置、またはC P U 1 2 B）は、メモリ2 8を直接アクセスできるが、認定なしではない。アクセス妥当性検査は、インターフェイス装置2 4のA V T論理回路

9 0によって実施されるように、それらのメモリ位置に書込まれるべきでない他のデータで良好なデータが誤って(erroneously) または不注意に(inadvertently) 上書きされることによりメモリ2 8の内容が汚染されることを防ぐべく動作する。同様に、アクセス妥当性検査は、間違ったメモリ位置を不注意に読取り、それにより読み取られるデータを要求しているエンティティまたはシステム素子に誤ったデータを供給する、アクセスに対する保護も供給する。これら及び同様な理由で、アクセス妥当性検査方法は、メモリ・アクセスが適切に行われること、即ち、適切な装置が適切なメモリ位置に書込むか、または適切なメモリ位置から読取ること、を確実にすべく供給される。入力メモリ要求（即ち、読取りまたは書込み）が確認されたならば、メモリ位置のアドレスは、要求を運んでいるメッセージ・パケットのアドレス・フィールドによって運ばれるように、メモリ・アドレスにA V T論理回路によって変換される。

【0 0 8 1】メモリ2 8へのアクセスは、6つ検査の全てを用いて、各インターフェイス装置2 4（図9）のA V T論理回路9 0によって確認される：（1）要求を運んでいるメッセージ・パケットのC R Cが誤りなしである、（2）メッセージ・パケットで識別された宛先（例えば、C P U 1 2 A）が受信機のそれである、（3）メッセージ・パケットで識別された要求のソースが正しいソースである、（4）シークされたアクセスの型がアクセスを要求しているソースに対して許容される、（5）アクセスがシークされるメモリ2 8の位置へのアクセスがソースに対して許容される、かつ（6）アクセスの転送サイズが所定のバウンド（上下限）内である。最初の検査は、上述したように、C R C論理回路チェッカ1 0 6によって、パケット受信機9 6で行われる。受信したメッセージ・パケットが不良C R C（または、以下に示すように、それが“このパケットは、不良”（T P B）指令記号でタグされる）を有することが見出されたならば、パケットは、廃棄され、かつアクセスは、否定される。メッセージ・パケット・ヘッダに含まれる宛先I Dは、パケットの宛先が正しい（即ち、C P Uによって受信されたならば、適切なC P U 1 2が宛先として指定される）ことを確実にすべく受信素子に割り当てられた宛先I Dに対して比較される。ミスマッチは、パケットが何らかで誤指向され、かつパケットが再び廃棄され、そして、もちろん、アクセスが再び否定されたことを表わす。

【0 0 8 2】残りの検査は、少なくともそのメモリがアクセスされる素子のメモリへのアクセスのある形が認められた各システム素子に対してアクセス妥当性検査（A V T）エントリ（図1 8）を、メモリ2 8において、保持することによってなされる。入力パケットのヘッダのアドレス・フィールドは、ソースI D（S o u r c e I D）において識別されたシステム素子に対するA V Tエ

ントリを含んでいるメモリ位置へのポインタとして用いられる。AVT論理回路は、どのアクセスがメッセージ・パケットの識別されたソースを許可されるかを決定するためにAVTエントリの妥当性検査情報を用いる。それゆえに、受信したメッセージ・パケットのソースIDフィールドは、パケットのクリエイタがCPU12のメモリ28へのアクセスを許可されるかどうかを決定するために用いられる。この検査は、特定のソースが特定の受信機の実効性へのアクセスを認められるべきかを決定するためにパケット・ヘッダのソースIDフィールドをAVTエントリ（ソースID）の部分と比較することを含む。シークされるアクセスの型（例えば、メモリの読取りまたは書込み）を識別している、パケットの型フィールドは、シークされるアクセスの型がメッセージ・パケットによって識別されたソースを許可されるかどうか、またはパケットが求められていない応答（誤りとして削除される）であるかどうかを決定すべく検査される。

【0083】そして、シークされたメモリ位置、及びあらゆる転送のサイズは、それらが特定のメッセージ・パケット・ソースも許可されるかどうかを見るために検査される。インターフェイス装置24aのアクセス妥当性検査機構、AVT論理回路88は、図16に詳細に示される。CPU12のメモリ空間へのアクセスをシークしている入力メッセージ・パケットは、パケット受信機96（図9）からAVT論理回路90のAVT入力レジスタ170へ転送されたそれらのヘッダの選択された部分を有する。従って、AVT入力レジスタ170は、ソースIDと、メモリ28に書込まれるかまたはそれから読取られるべきデータの量を識別している、レンジ（Len）フィールドと、AVT表エントリを含んでいるメモリ28のエントリを指し示している、アドレス（AVT頁#）と、AVTエントリが指し示すそのメモリ頁の中へのオフセットと、シークされたアクセスの型（Type）とを入力メッセージ・パケットから受信する。これらの値は、AVT入力レジスタ170のレジスタ・セグメント170a, 170b, . . . , 170eにそれぞれ含まれる。

【0084】AVT入力レジスタ170に含まれるAVT頁数フィールドは、妥当性検査に対して必要なAVTエントリのアドレスを生成すべく組合せ論理回路176によりそれがAVTベース・レジスタ174の内容と組み合わせられるAVTアドレス論理回路172に結合される。AVTベース・レジスタ174は、AVT表全体のメモリにおける開始アドレスを含む。生じたアドレスを用いて、AVTアドレス論理回路172は、次にAVTエントリ・レジスタ180の中に装填される、そのAVTエントリに対するメモリをアクセスする。AVTアドレス論理回路172は、AVT表に割り当てられたアドレス範囲内に入らないAVT頁数アドレスを検出する

AVTマスク・レジスタ175も含む。規則は、AVTマスク・レジスタ175のあるビット位置が0であるならば、AVT頁数アドレスの対応ビットも0でなければならない；そうでなければ、マスク検査論理回路177がマスク誤りを検出しかつメモリ28へのアクセスを否定すべく動作するというようなものである。AVTエントリ・アドレス生成及びマスク動作は、図17によく示されている。図17が図式で示すように、レジスタ・セグメント170cの20ビットAVT頁数値の高位8ビット部分は、AVT表エントリ・アドレスの高位部分（ビット16-31）を生成すべくAVTベース・レジスタ174の内容と合計される。同時に、レジスタ・セグメント170cからのAVT頁数アドレスの残りの（下位）12ビットは、AVTエントリ・アドレスの部分を直接形成する。AVTエントリは、4倍長語（quadword）量なので、それらは、4倍長語境界上に位置決めされる；それゆえに、AVTエントリ・アドレスの下位4ビットは、示すように、常にゼロである。

【0085】図17は、マスク動作も示す。AVT頁数アドレスの高位2バイトは、マスク・レジスタ175に含まれるマスクと比較される。0を含んでいるマスク・レジスタのビット位置が“1”を有する高位2バイトの対応ビット位置を検出したならば、マスク・レジスタは、メモリ28へのアクセスを否定する“マスク誤り（Mask Error）”信号をアサートし、かつプロセッサ20による作用に対して割り込み論理回路86

（図9）への割り込みを生成しかつポストする。マスク動作は、AVTエントリの表のサイズを変化させる。AVTマスク・レジスタ175の内容は、プロセッサ20にアクセス可能であり、プロセッサ20にAVTエントリ表のサイズを随意に選択させる。最大AVT表サイズは、あらゆる32ビットTNetアドレスの検証（及び変換）を許容する；即ち、最大サイズAVTエントリ表は、 20^{20} の異なる頁アドレスを検証しかつ変換することができる。最小サイズAVT表は、あらゆる24ビットTNetアドレス（即ち、高位8ビットがゼロであるようなTNetアドレス）の検証及び変換を許容する。最小AVT表は、 20^{12} の異なる頁アドレスを検証しかつ変換することができる。

【0086】従って、AVT表エントリが16バイトであるので、最大サイズAVT表は、専用メモリ空間の16メガバイトを必要とする。しかしながら、AVTマスク・レジスタ175及びAVTアドレス論理回路172の内容によって実行されるマスク動作は、AVTサイズをシステムのニーズにマッチさせる。非常に多数の外部素子（例えば、システム中のI/O装置の数は、大きい）を含む処理システム10は、広範囲のTNetアドレス、及び対応AVTエントリを必要とし、かつAVTエントリにメモリ28のかかなりの量のメモリ空間を確保し（割り当て）なければならない。逆に、小さな数の外

部素子を有する、小さな処理システム10は、小さなAVT表を用いることができよう。そのような大きなTNetアドレス要求を有さず、メモリ空間を節約する。従って、小さなシステムでは、高位ビットは、用いられない（または、より正確には、用いられるべきでない）。小さなAVT表が順番であるときには、TNetアドレスの高位ビットは、ゼロであるべきである；特定のシステムに対して範囲外であるTNetアドレスを有するAVT表エントリをシークする試みは、誤りである。マスク・レジスタ175の内容を用いて、そのような誤りを検出することはマスク論理回路の機能である。それゆえに、あらゆるCPU12（またはこの妥当性検査技術を用いている他のシステム素子）に対するときのAVT表拡張の許容可能サイズは、論理“ONE（1）”にセットされるビット位置によりマスク・レジスタ175の内容によって示される。論理“ZERO（0）”にセットされるマスク・レジスタ175のビット位置は、処理システム10の限界の外側の、存在しないTNetアドレスを表わす。許容可能TNet範囲外のTNetアドレスを有する受信パケットは、それがZEROであるべきところの論理ONEにセットされたビット位置を有する。AVTアドレス論理回路172は、この範囲外TNetアドレスを検出し、かつAVT誤り割込みを発行させる。

【0087】メモリ28で保守されることが必要なAVT表のサイズを変化することができるに加えて、上述したように、図17に示した技術は、また、可撓性を伴ってメモリ28にAVT表を位置決めさせるということが、いま当業者に明らかであろう。図17は、AVT表が 2^{17} （128K）の累乗境界上に位置決めすることができることを示す。各AVTエントリは、妥当性検査処理の間中にAVTエントリ・レジスタ180に保持されるときに図16に示したフィールドを含む128ビット4倍長語である。AVTエントリは、二つの基本フォーマットを有する：標準及び割込み。標準AVTエントリのフォーマットは、図18（及び、AVTエントリ・レジスタ180の内容を示すことによって、ある程度、図16にも）に示される；割込みフォーマットは、図20に示される。AVT論理回路90の説明を続ける前に、AVTエントリの意味及び内容の理解は、役に立つであろう。図18を参照すると、標準AVTエントリが52ビットPhysical Page Number（物理頁数）フィールドを含んで示されている。このフィールドの内容は、その内でアクセスがメッセージ・パケットの要求ソースを許容されるメモリ28の頁の物理アドレスを識別する。（一般に、各メモリ頁は、4K（4096）バイト・メモリ位置を含む。）Physical Page Numberフィールドの内容は、（妥当性検査をシークしているメッセージ・パケットのヘッダから引き出された）AVT入力レジスタ17

0に保持される12ビット・オフセット・フィールド170dの内容と連結される。結果は、妥当性検査が許可されるならば—データが書込まれるかまたは読取られるメモリ28内の位置の物理アドレス全体である。

【0088】アクセスが特定の4K頁の全メモリ位置に認められると同時に、アクセスは、また、その頁の部分だけに制限されうる。後者の制限を実施するために、AVTエントリは、アクセスが許可されるメモリ28の識別された頁内の上部及び下部バウンドを画定する二つの12ビット・フィールド（上部バウンド、下部バウンド；図18）を含む。特に、AVTエントリの下部バウンド・フィールドは、このAVT表エントリが適用される最低値を有するバイトのメモリ頁を有する（メモリ頁内の）オフセットを指定する。上部バウンド・フィールドは、このAVT表エントリが適用される最高アドレスを有するバイトのメモリ頁内のオフセットを指定する。この値（例えば、オフセット値170d+AVT入力レジスタ170のLenフィールド170bの内容）を渡すメモリ位置をアクセスする試みは、割込みを介してプロセッサにポストされる誤りを結果として生ずる。12ビット“Permissions（許可）”フィールドは、AVTエントリに対応している要求ソースに認められた許可を指定すべくAVTエントリに含まれる。Permissionsフィールドは、図19に示されており、あるPermissionsサブ・フィールド（E, PEX, PEY, I, C, W, R, 及びB）は、メモリ・アクセスへの以下の認定を識別する：

【0089】E:（Error Enable（誤りイネーブル））このAVTエントリを通して指向される誤ったアクセスは、このフィールドが二つの特定状態に一つ（例えば“ONE”）にセットされるならば（割込み論理回路に）報告される。

【0090】PEX:（Path Enable X（経路イネーブルX））この1ビット・フィールドの状態は、（全ての他の適用可能な許可も合致したならば）このAVTエントリを用いるべくゼロに等しいヘッダの“path（経路）”ビットで受信したメッセージ・パケットをイネーブルすべく“ONE”にセットされる。このビットが“ZERO”にセットされたならば、アクセスは、“x経路”（経路=0）にわたり受信したAVTエントリが適用されるメッセージ・パケットが否定される。否定は、割込み論理回路で割込みとしてログされ、かつEフィールドが誤り—報告をイネーブルする状態（“ONE”）にセットされたならば—プロセッサ20に報告される。

【0091】PEY:（Path Enable Y（経路イネーブルY））この1ビット・フィールドは、それがONE（1）にセットされた経路ビットで受信したメッセージ・パケットに適用されるということを除き、PEXフィールドと同じ方法で動作する。

【0092】I: (Interrupt (割込み))
このビットが(例えば、“ONE”に)セットされたならば、他のフィールド(上部バウンド、等)は、割込み書込みを処理しかつ割込みキューを管理するための新しい定義を得る。これは、割込み論理回路86の説明に関して以下により詳細に説明する。

【0093】C: (Cache Coherency (キャッシュ・コヒーレンシー))これは、どのようにメモリ28への書込み要求が処理されるかを指定すべく符号化された、2ビット・フィールドである。一つの状態にセットすると、要求書込み動作は、普通に処理される；第2の状態にセットすると、メモリのAVTエントリ・マップド領域の上部または下部バウンドに含まれた部分キャッシュ回線(fractional cache line)を有するアドレスを指定している書込み要求は、以下に説明する、割込みハンドラ250(図21)によって保守されるキャッシュ・コヒーレンシー・キューに書込まれる。これは、第3の状態にセットされた全キャッシュ回線アライメントを有さないメモリ28のユーザ・データ構造またはバッファ領域への書込み転送をCPU12に管理させ、このAVTエントリをアクセスする全ての書込み要求は、キャッシュ・コヒーレンシー・キューに書込まれる。第4の状態にセットすると、このAVTエントリによって参照される物理メモリ位置は、ハードウェア・コヒーレンシー機構を用いてアクセスされる。

【0094】W: (Write Access (書込みアクセス)) この1ビット・フィールドの状態は、Lower(下部)及びUpper(上部)Bound(バウンド)フィールドによって識別されるメモリ領域内で — 要求ソースに対してメモリへの書込みアクセスを認めるかまたは否定する。

【0095】R: (Read Access (読取りアクセス)) この1ビット・フィールドの状態は、指定したメモリ領域内で — 要求ソースが読取り動作に対してメモリへのアクセスを有するか否かを決定する。

【0096】B: (Barrier Access (バリア・アクセス)) この1ビット・フィールドの状態は、指定したメモリ領域内で — 要求ソースがバリア動作(以下に説明)に対してメモリへのアクセスを有するか否かを決定する。そして、AVTエントリの20ビット“SourceID”フィールドは、AVTエントリの許可情報が適用される特定のソースを識別する。図16に示されたAVT論理回路にここで戻ると、一度AVTエントリのアドレスが形成されると、エントリは、アクセスされかつAVT表エントリ・レジスタ180に一時的に記憶される。AVTエントリ・レジスタ180に含まれるように、Permissionsフィールドの内容は、アクセス論理回路184によりAVTエントリ・レジスタに保持されるTypeフィールドによって指定されたように、要求されるアクセスの型と比

較される。要求されたアクセスが許可されたものと一致しないならば、アクセスは、否定され、あつアクセス論理回路184は、ORゲート184及びANDゲート186を含んでいる誤り生成論理回路を介して生成されたAVTエントリ割込み信号をもたすべく誤り信号

(“No”)をアサートする。シークされたアクセスの型がPermissionsの一つでないならば、アクセスは、否定される。

【0097】(AVTエントリ・レジスタ180の“src ID”値として識別される)アクセスされたAVTエントリのSourceIDフィールドは、用いられるAVTエントリに対応するソースを指定し、かつ比較論理回路190によって要求メッセージ・パケットに含まれるSourceIDと比較される。再び、ミスマッチは、AVT Error Interrupt(誤り割込み)を生成させることを比較論理回路190に結果として生じ、そしてアクセスが否定される。同時に、AVTエントリのLower Boundフィールド(AVTエントリ・レジスタ180において“lwr bnd”として図16に示される)は、それがAVT入力レジスタ・セグメント170dにおけるOffset(オフセット)値と比較される論理回路194を比較すべく適用される。Offset値がAVTエントリのLower Boundフィールドに含まれたものよりも小さく、アクセスが許可された頁部分の外側であるということを示しているならば、コンパレータ(比較器)194は、ORゲート184及びANDゲート186を介して、AVT誤り割込みを生成する、信号を始動し、メモリ28へのアクセスを否定する。

【0098】同様に、比較論理回路196は、(書込み動作が要求されたならば)書込まれるべきデータの量が、(エントリのLower及びUpper Boundフィールドによって画定されるように)要求ソースに割り当てるメモリ空間の量を超えるかどうかを決定すべく — 加算器論理回路200から — Upper Boundフィールド(AVTエントリ・レジスタ180の“upr bnd”)をLenフィールド(即ち、書込まれるべきデータ・バイトの数)及びOffsetの合計と比較する。アクセス要求が確認されたならば、AVTエントリ・レジスタ180のPhysical Page Number(phys pg #)内容は、アクセスが行われるメモリ位置をアドレス指定すべく、図17に関して上述したように、AVT入力レジスタ170からのOffsetと一緒に、用いられる。システム10の素子(例えば、装置17とCPU12；図1、図2及び図3)間のメッセージ・パケットの通信は、ことのほか、アクティビティを要求するか、またはアクティビティを報告するか、或いは誤りの発生を知らせるべく割込みを分配する、新規な方法にも採り入れられる。それゆえに、割込みメッセージ・パケット送付

は、他の素子間通信と同じ方法でTNetネットワーク・システムを用い、かつ3つのステージ進行を含む：

(1) ソース素子からの割込みメッセージ・パケットの生成及びタスク指名(ディスパッチ)；(2) その宛先へのTNetネットワークを通る割込みメッセージ・パケットの伝播；(3) 宛先での作用に対する解釈及び“ポスティング”。あらゆるシステム素子は、割込みメッセージ・パケットの受信者でありうる。宛先がCPUであるならば、割込みメッセージ・パケットは、実質的に、メッセージ・パケット・ヘッダのDestination IDフィールド(図5(b))がCPUを識別し、かつAddressフィールドがどのように割込みメッセージ・パケットが処理されるべきかの命令を含んでいるAVTエントリ(Interrupt Descriptor)を選択するような標準“書き込み”要求である。

【0099】割込みアクティビティの始動するための許可は、また、AVT論理回路88によって確認されなければならない。それゆえに、受信したメッセージ・パケットは、割込みを説明する割込みデータを含む。その割込みデータは、割込みが受信されかつ“ポスト”され、そしてプロセッサ20によってサービスする準備ができていることを示すべくプロセッサ20に供給する信号で、メモリ28の特定のキュー(割込みキュー)に書込まれるべきである。割込みキューは、特定のメモリ位置にあるので、プロセッサは、必要なときに割込みデータを得ることができる。割込みに対するAVT割込みエントリは、二つの型の一つでありうる：マルチエントリ・キュー型割込み、または単一エントリ・キュー型割込み。AVT割込みエントリの両方の型に対するフォーマットは、基本的に同じであり、そのフォーマットは、図20に示されている。マルチエントリ・キュー型割込みに対するAVT割込みエントリは、割込みを送付すべく最初に構成されたか、またはルータ14或いは受信CPU(例えば、不良CRC)によって検出された例外により途中で割込みになったかのいずれかである受信したメッセージ・パケットに対して用いられる。これらのエントリは、上述したのとほぼ同じ方法でメッセージ・パケットを確認すべくAVT論理回路90によって用いられ、かつ割込みメッセージ・パケットのヘッダ、及び付随しているデータが記憶されるメモリ28の円形キューを識別すべく割込み論理回路86(図9及び図21)によって用いられる。更に、割込み論理回路86は、信号受信及び/又はマルチエントリ割込みの生成に対して割込みまたは“原因”レジスタ280(図21：以下により完全に説明される)にビットをセットする。

【0100】単一エントリ・キュー型割込みに対するAVT割込みエントリは、AVT割込みエントリが記憶に対してメッセージ・パケット情報を指向する割込みデータ構造が、メモリ28における(変更可能であるが)固

定された位置であるということを除き、実質的に同じような方法で動作する。両方のAVT割込みエントリ型(マルチエントリ及び単一エントリ割込み)は、図20に示した4倍長語(128ビット)フォーマットを有する。AVT割込みエントリの64ビット・セグメント(“Queue Base Addr(キュー・ベース加算器)”)は、割込みデータが書込まれる割込みキューのメモリ28の位置へのポインタとして用いられる。割込みキューは、割込みをサービスするときに、割込みデータが割込み論理回路86によりFIFOキューのテール(tail)で受信されかつ挿入され、かつプロセッサ20によりキューのヘッド(head)から抽出されるようにFIFOの形で構成される。また、AVT割込みエントリは、ソースID情報を含んでいる20ビット・セグメント(“Source ID”)も含み、割込み処理による注意をシークしている外部装置を識別する。AVT割込みエントリのソースID情報が、コンパレータ190(図16)によって実行される比較によって決定されるように、入力メッセージ・パケット(Source；図5(b))のヘッダに含まれたものとマッチしないならば、割込みキューへのアクセスは、否定され、かつAVT誤り割込みが生成される。

【0101】AVT割込みエントリの12ビット“Permissions”セグメントは、標準AVTエントリに関して上述したのと同じ許可情報を含む。しかしながら、一般に、割込みを送付するメッセージ・パケットは、書き込み要求として構成され、それがメモリ28に運ぶ割込みデータを書込むべくシークする。それゆえに、適切なAVT割込みエントリは、割込みデータをメモリ28に書込ませるべくセットされたWrite Access(書き込みアクセス)ビット(W)を有する。Permissionsフィールドの割込みビット(I)は、セットされたときに、割込みメッセージ・パケットを確認しかつ処理するためのものとしてAVT割込みエントリを識別する。そして、AVT割込みエントリの4つの、1バイト・セグメント(“c”, “q”, “l”, 及び“z”)は、(1)プロセッサ20にセットされた割込みレベルを決定するために用いられる割込みの“class(クラス)”(以下により完全に説明される)；(2)、その内容が(Queue Base Address(キュー・ベース・アドレス)フィールドによって識別された)特定のキューのどこに割込みデータが書込まれるべきであるかということを示す、レジスタを、分かるように、選択するために用いられるキュー数；(3)そこに記憶することができる倍長語の数による各キュー位置で利用可能な記憶装置のサイズまたは量；及び(4)キューのどこにデータが書込まれるかを識別するために用いられるキュー・テール・カウンタにおけるビットの数、をそれぞれ識別する。Queue Base Addr、とc, q, l, 及びzセグメン

トは、メモリ28の位置をポイントすべく割込み論理回路86によって用いられる。割込み論理回路86は、それぞれが割込みデータを挿入することができる4つのキューの一つをポイントする4つの“テール”・カウンタを含む。4つのカウンタの特定の一つは、AVT割込みエントリのqセグメントの内容によって選択される。そこから割込みエントリが引き出される点である、キューの他端は、4つの“ヘッド”・カウンタの一つによって識別される。ヘッド及びテール・カウンタの(ビット数による)サイズは、以下に示す表2に指定されたように、9により負にバイアスされた、zサブ・フィールドによって指定される。キュー・テール・カウンタ・サイズは、テール・ポイントがゼロの値にラップバック(wrap back)するときを決定するために用いられる。各エントリのサイズによって分割された語の数(バイト)は、キュー・エントリの数を与える。最短キューは、ほんの32エントリ(128バイト・エントリによって分割された4k Bキュー)を有して、4k Bを必要とすると同時に、最長キューは、32, 768エントリ(エントリ当たり16バイトによって分割された512k B)と同じ数だけ有することができる。

【0102】

【表2】表 2

Z	解	積
0	512 倍長語	(4K バイト)
1	1K 倍長語	(8K バイト)
2	2K 倍長語	(16K バイト)
3	4K 倍長語	(32K バイト)
4	8K 倍長語	(64K バイト)
5	16K 倍長語	(128K バイト)
6	32K 倍長語	(256K バイト)
7	64K 倍長語	(512K バイト)

8-15 使用せず

【0103】割込みキューの各割込みエントリのサイズは、以下の表3に示した方法で、4倍長語で、1フィールドによって指定される。

【0104】

【表3】表 3

1	解	積
0	1 4倍長語	(16 バイト)
1	2 4倍長語	(32 バイト)
2	4 4倍長語	(64 バイト)
3	8 4倍長語	(128 バイト)

4-15 使用せず

【0105】- 割込み処理：上記したように、本発明の新規の特徴(機能)は、サービスをするためにCPU12に割込みを送付すべくTNetネットワーク・メッセージ送付能力を用いる能力(機能)である。例えば、I/O装置は、トランザクションを送付したメッセージ・パケットにおける不適当なアドレス、またはCRC誤

りを有するメッセージ・パケットの受信、または受信メッセージ・パケットが受信者を識別しなかった宛先アドレスを有していたということに気付く(注目する)ことのような多数の理由によりCPUによって発行される読取りまたは書込みトランザクションを終了することができないであろう。I/O装置、またI/Oインターフェイス素子によって気付かれた(注目された)、これら及び他の誤り、例外、及び不規則性は、CPUの介入を必要とする条件(状態)になることができる。従来のシステムでは、そのような条件(状態)は、割込みの対象事項である；そこで、それらは、そのような割込みが送付されないということを除き、過去におけるように - 割込み条件(状態)についての情報を殆どまたは全く有さなく、かつそのような目的のためにだけ確保された信号回線によって - しかしシステムのI/O素子に利用可能なメッセージ・システムを通して、ここにある。この特徴(機能)は、余分な信号回線に対する必要性を低減する(他の使用に対して信号回線空間が利用可能であるものを許容する)だけでなく、CPUが調査を実行するために時間を費やさなくてもよいように割込みの原因になったものに対するより多くの情報を供給することができるファシリティを供給する。

【0106】この特徴(機能)により、メモリ28に書込まれるべき割込みタスク指名を含んでいる、入力メッセージ・パケットは、妥当性検査のためにAVT論理回路90(図16)にまず渡される。また、AVT論理回路90は、メッセージ・パケットが正規のI/O書込み要求、割込み、あるいは禁止されているメモリ28への誤アクセスであるか否かを決定する。AVT論理回路90のAVTベース・レジスタ174の内容は、標準メッセージ・パケットに対して上述したのと同じような方法で主メモリにおけるAVT割込みエントリに対するポイントを生成するためにAVT入力レジスタ170に含まれる頁数フィールド170c(図16)を伴って用いられる(例えば、データを読取るかまたは書込むためにメモリ28へのアクセスをシークすること)。形成されたアドレスによってそのように識別されたAVTエントリは、メモリ28からアクセスされかつインターフェイス装置24(図9)の割込み論理回路86による使用のためにAVTエントリ・レジスタ180にセットされ、図21に詳細に示される。一度その割込み情報を運んでいるメッセージ・パケットがAVT論理回路90によってクリアされたならば、割込み情報を処理する役割を果たすのは、割込み論理回路86である。割込み論理回路86は、4つのキュー・テール・レジスタ256の内容を受信し、かつそれらの間で選択する、マルチプレクサ(MUX)252を含んで、図21に示されている。同様に、MUX254は、4つのキュー・ヘッド・レジスタ262の内容を受信し、かつそれらの間で選択する。各MUX252, 254の選択入力、AVTエントリ

・レジスタ180に保持された(割込みメッセージ・パケットに対応している)検索AVTエントリの“q”セグメントの内容を受信すべく結合される。使用すべくキュー・レジスタ256, 262の各グループからの一つを選択するために用いられるのは、q値である。

【0107】数は、ここでは、以下に説明する理由により4つに制限されるが、割込みデータの記憶を処理するためにメモリにセット・アップされたあらゆる数のキューが存在しうる。各そのようなキューのメモリ28内の位置は、アクセスされたAVTエントリのキュー・ベース・アドレス値によって指定され、かつエントリ・レジスタ180(図16の“phys pg#”; 図21の“ベース”)に保持される。4つのキュー・テール・レジスタ256の内容は、それぞれが特定のキューの中にオフセットを形成してキュー・ベース・アドレス値によって指定される。選択したキュー・テール・レジスタ256の内容は、加算器258によりキュー・ベース・アドレスと組合わされ、そこで割込みデータが書込まれる指定されたキューの中にエントリ・ポイントを形成する。さらに多くのまたは少ないキューを維持(保守)することができるということは、当業者には、明らかなことであるが、4つのキュー・ヘッド及びテール・レジスタ262, 256は、4つのキューを処理することだけに割込み論理回路86を制限する。レジスタ256は、特定のキューの“テール”の位置を指定し、次の受信割込みデータが配置されるキュー・エントリを指し示す。4つのキュー・ヘッド・レジスタ262は、特定のキューの他端を指定する。

【0108】キュー・ベース・アドレスと選択されたテール・キュー・レジスタ256の内容との組合せから生じられたアドレスは、4倍長語(16バイト)境界上に位置合わせすべく形成されるのが好ましい。これは、キュー・エントリ・アドレスの下位4ビットを0に強要することによって達成される。キュー・エントリ・アドレスの形成は、図22に図式的に示され、選択されたテール・レジスタ256の15ビット内容の高位7ビットが、AVTエントリ・レジスタ180に含まれるキュー・ベース・アドレス・フィールドのビット位置12-31に付加されることを示している；この合計の結果は、キュー・エントリ・アドレスの高位20ビット(ビット位置12-31)を形成する。選択されたテール・レジスタ256の下位8ビット内容は、キュー・エントリ・アドレスの下位ビット位置4-11として直接用いられる。上述したように、キュー・エントリ・アドレスの下位4ビット(ビット位置0-3)は、所望の位置合わせに対して全てゼロに強要される。纏めると、割込みを含んでいるメッセージ・パケットは、それらが、実質的に、受信CPU12のメモリ28にデータを書込むための要求であるので、他のメッセージ・パケットと同じ方法で最初処理され、その要求は、AVT論理回路90に

よって確認されなければならない。それゆえに、メッセージ・パケットからの情報は、AVT入力レジスタ170及びAVTエントリを位置決めしかつメモリ28からアクセスするために用いられる部分(フィールド170c及び170d)にセットされる。AVTエントリは、メッセージ・パケットが割込み処理に対する適当な情報を含むならば、AVTエントリ・レジスタ180にセットされる割込みAVTエントリでありかつ割込みを確認(オーセンチケート)するために用いられ、そして、割込み論理回路86を用いて、AVTエントリに含まれるベース・アドレス情報によって指定された4つの円形キューの一つに割込みデータを記憶する。プロセッサ20は、通知され、かつ割込みが処理されるか否か、及びどのように処理されるということは、それら次第である。

【0109】キューに割込みメッセージ・パケット・データを記憶して、アドレス指定は、次のメッセージ・パケットの割込みデータの受信を見越して更新されなければならない。割込みデータが選択キューに書込まれた後、AVT表エントリ・レジスタ180に含まれる

“1”フィールドの内容は、組合せ(combiner)回路270により選択テール・キュー・レジスタ256と組合わされ、その出力は、次の割込みメッセージ・パケットの割込みデータが記憶されるところで新しいオフセットをキューに変える(turn into)べく“mod z”回路273によって処理される。その新しいオフセット値は、選択テール・キュー・レジスタ256に戻される。同時に、組合せ回路270の出力は、比較回路272に供給される。割込み問合わせは、zのモジュール・サイズを有するファッションで環状であるべく構成される。mod z回路は、環状性を維持する出力を生成する。テール・キュー・ポイントがキューにおけるネスト・エントリ・ポイントを識別し、かつヘッド・ポイントがキューにどのくらいのルームが残っているのか、対応テール・ポイントに関して、識別するので、これら二つの値が等しいならば、キューは、充満(いっぱい)である。それゆえに、(選択されたヘッド・キュー・レジスタ262によって供給された)ヘッド・ポイントを最後のエントリの結果として生成されたテール・ポイントと比較することによってこの決定を行うのは、比較回路272である。キューに対してヘッド及びテール・ポイントがいま等しいならば、比較回路272は、それ自身が割込み信号である、“Queue Full(キュー充満)”警告信号を発行する。Queue Full警告信号は、事項(matter)が適格に処理されないならば、キューがいっぱいであったならば、追加割込みメッセージが廃棄されるので、遅く受信した割込みデータが失われうるといふ、警告としてプロセッサ装置20に運ばれる“固有(intrinsic)”の割込みになる。

【0110】入力メッセージ・パケット割込みは、割込みレジスタ280の多数のビット位置の一つをまずセッ

トすることによって割込みをプロセッサ20にポストさせる。マルチエントリ・キュー型割込みは、プロセッサ20にポストするために割込みレジスタ280aにセットされる；単一エントリ・キュー割込みは、割込みレジスタ280bを用いる。どのビットがセットされたかは、AVT入力レジスタ180に保持されたAVTエントリのクラス・フィールド(c)に依存する。

【0111】まずマルチエントリ・キュー型割込みを考慮すると、マルチエントリ・キュー型割込みが決定された後すぐに、インターフェイス装置は、復号回路283に印加(供給)される対応割込み信号(I1)をアサートする。復号回路283は、セットすべきレジスタ280aのレジスタ位置を決定すべくAVTエントリ・レジスタ180からクラス(c)値を受信しかつ復号し、それによりプロセッサ20に受信割込みに関する先進の情報、即ち、(1)ポストされた割込みの型、及び(2)その割込みのクラス、を供給する、同様に、単一エントリ・キュー割込みは、受信したときに、セットすべきレジスタ280bのビット部分を決定すべくクラス(c)値を受信しかつ復号する、復号論理回路287に対応割込み信号(I2)をアサートさせかつ印加させる。

【0112】テール及びヘッド・キュー・レジスタ256, 262は、マルチプレクサ(MUXs)276, 274の別のペアにも結合される。更新レジスタ278の内容は、比較回路279によって互いに比較されるレジスタ256, 262の対応ペアを選択する。更新レジスタは、比較のためのレジスタ・ペアを選択するためにプロセッサ20によって書込み可能である。二つの選択されたレジスタ256, 262の内容が等しく、対応キューが空であるということを示していることがわかったならば、対応割込みレジスタは、クリアされる。クラス・レジスタ281は、クリアされることが必要な割込みレジスタ280aの割込みビットを(クラスにより)選択する。ちょっと脇道にそれると、プロセッサ20に係わる二つの割込みの基本型が存在する：メッセージ・パケットによりCPU12に伝達される割込みと、“固有”割込みと呼ばれる、CPU12自身によって生成される割込みである。固有割込みは、割込み論理回路86の比較回路272によって生成されるキュー充満警告信号のような内部的に検出された誤りの結果として生ずる。しかしながら、それらは、最初に割込みパケットとして送られなかったメッセージ・パケットを受信するときに気付く例外も含みうる。そのようなメッセージ・パケット割込みは、メッセージ・パケットが不良指令記号を有しているものとして検出されたこと、または受信メッセージ・パケットが不要CRCを有する(かまたは以下に説明するように、TPB識別子でタグされる)ことを見出した結果として生ずる誤り。これらの固有割込みは、固有割込みがマルチエントリ及び単一エントリ割込みが

割込みレジスタ180a, 180bのビット位置をセットすることによってポストされるのと同じ方法でポストされるところに固有レジスタ280cの特定ビット位置をもたらす。更に、メモリ28に維持されるAVT表は、固有AVT割込みのために確保された第1の数のエントリを有する。固有割込み信号が固有割込みレジスタ180cをセットすべく生成されたときには、それは、割込みをアクセスさせかつAVT論理回路90のAVTエントリ・レジスタ180に装填させた例外に対応しているAVTエントリをもたらす。それ以後、割込みは、メッセージ・パケット送信型割込みとして同じ方法で処理される。

【0113】ビット毎に基づき、割込みレジスタ280a, 280b, 及び280cの各々に関連するのは、それぞれ対応マスク・レジスタ282a, 282b, 及び282cである。割込みレジスタ280(例えば、280a)の各ビット位置は、マスク・レジスタ282(例えば、282a)における対応ビット位置を有する。マスク・レジスタ282の特定ビットがセットされたときには、関連割込みの認識は、抑制される。割込みレジスタ280の内容は、マスク・レジスタ282の内容によって渡されたならば、それらが7つの割込み“ポスティング(postings)”(信号)に組合わされる、複数のORゲートを含んでいる、組合せ論理回路286に結合される。組合せ論理回路286は、7つの割込みポスティングをラッチ288に結合し、ラッチからそれらは、ポスティングを受信しかつ保持するポインタ割込みレジスタを有するプロセッサ20(20a, 及び20b)に結合される。加えて、レジスタ288の内容は、比較回路289に印加(供給)され、かつレジスタ288の入力と(レジスタ288を装填する各クロックの前に)比較される。割込みにおける変化(割込みがサービスされ、かつそのポスティングがプロセッサ20によって削除されたか、または新しい割込みがポストされたかのいずれか)を示している、相違が存在するならば、“CHANGE(変化)”信号が、割込みポスティング変化が発生し、かつ変化をプロセッサ20に伝達すべきであることをそれに報告すべくプロセッサ・インターフェイス60に発行される。

【0114】AVTエントリ・レジスタ180は、TAG及び有効ビットを備えた単一回線キャッシュのよう動作すべく構成されるのが好ましい。TAGは、システム・メモリ28からAVTエントリを調べる(ルックアップ)ために用いられるTNe tアドレスの部分からなる。通常動作では、TAGが入力パケットのTNe tアドレスとマッチしないならば、正しいAVTエントリがシステム・メモリ28から読取られかつAVTエントリ・レジスタ206に読取られて、古いAVTエントリを置換する。当業者は、ちょっと挙げてみると、セットアソシエイティブ、完全一関連(fully-associate)、また

は直接マッピング型(direct-mapped)のような他のキャッシュ編成が可能であるということを認識するであろう。

— コヒーレンシー：キャッシュ・メモリ用いるデータ処理システムは、コヒーレンシーの問題を長く認識していた：キャッシュまたは主メモリへのアクセスが新鮮でないデータを決して戻さないか、または良好(最新)なデータを上書きしないことを確実にすること。この問題に対する多数の解決策が存在するが、その多くが大規模かつ複雑なハードウェアを使用する。コヒーレンシー問題は、データが外部(CPUに対して)I/Oまたは別のCPU12からメモリに書込まれるとき、または、システム10のコンテキストにおけるように(例えば、図4)、データがCPU12BによってCPU12Aのメモリ28に書込まれるときにも生ずる。一つの解決策は、入力データがバッファのバウンドがキャッシュ・ブロック境界に位置合わせされるようにメモリ・バッファに書込まれることを確実にすることである。しかしながら、この解決策は、入力データに対して用いられるキャッシュ・ブロックの妥当性を検査するためにソフトウェア・スキームと一緒に用いられるときにだけ応用(アプリケーション)を見出し、出力データに用いるキャッシュ・ブロックの書戻しを強要する。

【0115】それゆえに、(I/O、または別のCPU12からの)入力読取り要求、及び出力読取り及び書込み要求に適するコヒーレンシー問題のソフトウェア管理に対する従来技術が存在する。しかしながら、従来技術は、キャッシュ・ブロック境界上に位置合わせされていないメモリ28のI/Oバッファへの入力書込み要求を管理するのに向いていない。しかしながら、キャッシュ・ブロック境界上にメモリのI/Oバッファの位置合わせを要求することは、より可機性が少ないシステム、及び既存の(オペレーティング・システム)ソフトウェアと非互換性でありうるシステムを結果として生ずる。従って、本発明の割込み機構は、そのバッファの境界がキャッシュ・ブロックの境界に位置合わせされるか否かということを考慮することなくデータ・バッファをメモリに位置決めさせるようにコヒーレンシーを確立するために用いられる。この関連において、入力パケットのソースがアクセスを許容されるメモリ28の領域の上部及び下部境界(upr bnd, lwr bnd)を画定しているAVT表Entryレジスタ180のフィールド(図16)は、境界クロッシング(Bdry Xing)検査装置219に印加(供給)される。境界検査装置219は、それでCPU12が動作すべく構成されるキャッシュ・ブロックのサイズの表示、AVT Entryレジスタ180に保持されるAVTエントリのPermissionsフィールドからのコヒーレンシー・ビット("c[1:0]"),及びAVT入力レジスタ170からのヘッダ情報のLenフィールドも受信する。Bdry Xing装置は、入力パケットのデータ

が 上に位置合わせされていないかを決定し、かつコヒーレンシー・ビット("c[1:0]")が適切にセットされたならば、データ及びそのデータを含んでいるパケットのヘッダを記憶するために特別のコヒーレンシー・キューを指し示すべく用いられる割込みエントリのアドレスのフェッチを強要する。

【0116】図42をここでちょっと参照すると、CPU12のメモリ28(図4)によって実施されるメモリ空間の部分28'が示されている。図42が更に示すように、3つのキャッシュ境界CB_a, CB_b, 及びCB_cは、メモリ部分28'と一緒に含まれ、二つのキャッシュ・ブロックC_BLK_a及びC_BLK_bを画定する。書込み要求メッセージ・パケットが受信され(例えば、別のCPU12、またはI/O装置から)、かつクロスハッチング(格子縞)で示される、そのメッセージ・パケットに含まれるデータがメモリ部分28'を含むメモリ28の領域に書込まれるものであると想定する。事実、書込まれるべきデータは、キャッシュ・ブロックC_BLK_aをほんの部分的に上書きするが、キャッシュ・ブロックC_BLK_b、及び他のキャッシュ・ブロックを完全に上書きする。書込まれるCPU12のキャッシュ22がキャッシュ・ブロックC_BLK_b、またはキャッシュ・ブロックC_BLK_a以外の他のキャッシュ・ブロック(または、キャッシュ境界上に位置合わせされていないならば、入力データの他端を含んでいるキャッシュ・ブロック)を含むならば、ブロックは、“無効”とマークすることができ、メモリに書戻されかつ新しく受信したデータを上書きすることからそれを防ぐ。

【0117】しかしながら、キャッシュ22がキャッシュ・ブロックC_BLK_aを含むならば、AVT90の("c"がPermissionsフィールドにセットされることによりイネーブルされたならば;図16及び図19を参照)境界クロッシング論理回路219は、キャッシュ・エントリを部分的に無効にするI/Oパケットを検出し、かつコヒーレンシー割込みを強要することが必要である。これは、特別割込みキャッシュへのポインタを含んでいる、割込み記述子のフェッチを結果として生じ、かつ全入力TNet要求パケットは、キューに書込まれる。同時に、割込みは、入力データの一部分が特別のキューに装填されるということをプロセッサ20に知らせるためにキュー型割込みレジスタ280に書込まれる。

【0118】要するに、入力パケットがメモリ28に書込まれるべきデータを有するならば、境界クロッシング論理回路219は、データが書込まれるバッファの境界がキャッシュ境界と位置合わせされるかどうかを見る(調べる)べく検査する。そうであるならば、データは、指示(指導)されるように書込まれる。そうでなければ、パケット(ヘッダ及びデータの両方)は、特別の

キューに書込まれ、かつプロセッサは、上述した固有割込み処理によってそのように知らされる。次に、プロセッサは、特別のキューからキャッシュ22にデータを移動し、よいデータが上書きされないかそもなくば失われず、かつキャッシュ22とメモリ28の間のコヒーレンシーが保存されることを確実にすべく後でキャッシュをメモリ28に書込みうる。

【0119】— ブロック転送エンジン(BTE)：プロセッサ20は、CPU12Aの外部素子に情報を直接伝達(例えば、送る)することを禁止されているので、CPUのインターフェイス24a(図9)のBTE88は、情報送信の間接方法に対して供給される。BTE88は、情報のブロックを転送すべく全プロセッサ始動型(all processor initiated) I/Oトラフィックを実施するために用いる機構である。BTE88は、TNet packets規定によった許容される最大値、現在は64バイト、までの長さを有する読取り及び書込みパケットの生成を許容する。BTE88は、一つが他よりも高い優先度を与えられる、二つの“仮想”チャンネルを供給する。図23を参照する、BTE88は、それらの内容が(インターフェイス装置24aの；図9)MUX306に結合されかつメモリ・コントローラ26(図23に図示せず)を介してシステム・メモリ28をアクセスするために用いられる二つのBTEレジスタ300、302を含んで示されている。レジスタ300、302の部分は、CPU12A(図4)のメモリ28におけるBTEデータ構造304の始まりへのポインタ(例えば、BTEアドレス0及びBTEアドレス1)を含む。プロセッサ20は、CPU12Aの外部の一つまたは別の素子(例えば、CPU12BまたはI/Oパケット・インターフェイス16のI/O装置17、18のどちらか)に情報が送られるかまたはそれから情報が検索される度にメモリ28にデータ構造304を書込む。各データ構造は、4倍長語境界上で始まることを必要とし、かつBTEレジスタ300、302は、プロセッサ20によってのみ書込み可能である。プロセッサがBTEレジスタ300、302の一つに書込むときには、それは、BTE状態マシン307によって制御される、BTE処理を開始すべく動作する、クリア状態に要求ビット(rc0, rc1)をセットする語でそのようにする。

【0120】BTEレジスタ300、302は、タイムアウト/NAK誤り表示を報告する誤りビット(e0, e1)も含む。誤りビットは、対応BTEレジスタが書込まれるときにクリアされる。誤り原因(ec)ビットは、タイムアウトとNAKsとを差分する。情報が外部装置にプロセッサ20によって転送されるときに、データ構造304のデータ・バッファ部分304bは、転送されるべき情報を保持する。外部装置からの情報がプロセッサ20によって受信されたときには、データ・バッファ部分304bは、読取り応答情報を保持す

べくターゲットされた位置である。データ構造304の始まり、プロセッサ20によって書込まれた部分304aは、送られるパケットを受信する外部素子を識別する、情報フィールド(Dest)を含む。部分304aは、所望の動作(例えば、読取りまたは書込み情報)を記述する情報フィールド(TYPE)、書込まれるかまたは要求されるデータのバイト数を記載している、長さ情報フィールド(Len)、及び所望のデータが位置決めされる外部素子(Dest)の場所、または送信されたデータが書込まれるべき場所を識別している、アドレス情報フィールド(Address)も含む。この情報は、図5(a)～図5(d)及び図6～図8に示す形にパケットをアSEMBLすべくパケット送信機装置120(図9)によって用いられる。

【0121】データ構造部分304aにおけるアドレス情報にすぐ続くのは、データ・バッファ部分304bが位置決めされるメモリのアドレスを含んでいる語(Local Buffer Ptr)である。次に、チェイン・ポインタ、要求が終了したことを示すエンドオーバーリスト(e1)フラグ、終了インディケータ上の割込み(ic)及びチェックサム(cs)要求を含んでいる語が、その後すぐに続く。一つのデータ構造304は、最大64バッファ長まで外部素子(例えば、I/O記憶装置)に移動されるべくメモリのデータの各部分に対して用いられる。BTEは、データの64バイト・セグメントに対して、各要求構造に応じて、メモリ28を逐次的にアクセスすべく動作し、各セグメントに対してメッセージ・パケットを形成し、かつ進行してそのメッセージ・パケットを送る。チェイン・ポインタは、エンドオーバーリスト・ビット(e1)がセットされない限り、データの別の64バイトに対する次のデータ構造にBTEを指向して、動作を終わらせる。データが多数の外部素子へ送られるべきであるならば、各異なる素子は、セットアップされるべくそれ自身のデータ構造(64バイト以上が送られるならば複数のデータ構造)に要求する。次に、これら個別のデータ構造は、要求構造のチェイン・ポインタ・フィールドに含まれるチェイン・ポインタを用いて、チェインで繋がれる。チェイン・ポインタ・フィールドは、後続データ構造に対するBTEレジスタの内容として用いられる。例えば、メモリ28の大きなブロックのデータがNの異なる外部素子に送られるべきであるならば、データ構造は、BTE論理回路88が送られるべきデータを見出すことができるメモリ28の場所を識別している各データ構造でNの外部素子のそれぞれに対するメモリに書込まれる。各素子に対するデータは、BTE論理回路88によってアクセスされ、メッセージ・パケットがデータを含んで形成され、かつそれらが適切にTNetに送られるパケット送信機120に伝達される。次に、データ構造に含まれるチェイン・ポインタは、別のデータ構造にチェインで繋がれるこ

とが必要であるならば、アクセスされかつ作用を始動した適切なBTEレジスタ300、302に書込まれて、要求パケットを受信すべく次の素子のための次の構造に対するアドレスを供給する。

【0122】エンドオブリスト(e1)ビットは、セットされたときには、チェーンの終りを示し、かつBTE処理を停止する。割込み終了(ic)ビットは、セットされたときには、インターフェイス装置24aに、前のBTE送信パケット(チェーン・ポインタによって示されたものではない)の終了を示すために割込みレジスタ280(図21)にビットをセットする割込み(BTECmp)をアサートさせる。割込みタイムアウト(it)ビットは、セットされたときには、インターフェイス装置24aに、アクセスの肯定応答がタイムアウト(即ち、要求タイマ(図示省略)がタイムアウト信号を発行して、予期応答を適切な時間内に受け取らなかったことを示す)するか、またはNAK応答を導き出す(要求のターゲットが要求を処理できないということを示している)ならば、プロセッサ20に対する割込み信号をアサートさせる。そして、チェック・サム(cs)ビットがセットされたならば、外部素子に書込まれるべきデータは、チェック・サム量を生ずるべくBTE88(インターフェイス24a;図9)のチェック・サム発生器(図示省略)を通過させる。生成されたチェック・サムは、メモリに書込まれ、それ自身のパケットに続いて位置決めさかつチェック・サムが形成されたデータを含んでいるメッセージ・パケットの宛先に送られうる。

【0123】纏めると、CPU12Aのプロセッサ20が外部装置にデータを送ることを望むときには、それらは、データ構造の部分304aに識別子情報、バッファ部分304bにデータを含んでいる、データ構造304をメモリ28に書込む。次に、プロセッサ20は、データの優先度を決定し、かつデータ構造304(即ち、ヘッダ部分304a)を見出すことができるメモリ28のアドレスを有するBTEレジスタ300、302を書込み、同時にBTEレジスタ300、302の要求終了ビット(rc1)をクリアして、BTE動作を、BTE状態マシン306の制御下で開始させる。Dest, TYPE, Len, 及び部分304aからのアドレス情報は、メモリ28からアクセスされかつ適切なパケット形に配置されるパケット送信機120に伝達される。データ構造304が、転送が書込み動作であることを指定するならば、局所バッファ・ポインタは、アクセスされかつデータ・バッファ部分304bを位置決めするために用いられる。次に、データは、アクセスされ、パケット送信機120に伝達され、ヘッダ及びアドレス情報と一緒にパケット化され、かつ送られる。データ構造304が読取り要求(例えば、プロセッサ20が外部装置I/O装置またはCPU12のいずれかからのデータをシークする)を示すならば、Len及びLocal

Buffer Ptr情報は、(外部素子から要求が行われたところに)読取り応答パケットが戻されたときに、メモリ28への書込み要求を生成するために用いられる。データは、局所メモリ書込み動作が実行されるまでパケット受信機100(図9)の入力パケット・バッファ110に保持される。

【0124】外部装置へのプロセッサ生成型読取り要求に対する応答は、AVT表論理回路146によって処理されない。それよりも、プロセッサ20がBTEデータ構造をセットアップしたときに、トランザクション・シーケンス番号(TSN)は、要求が割り当てられ、かつ上述したHAC型パケット(図6～図8)である、BTE88によって形成されかつ送られるメッセージ・パケットのヘッダ・フィールドに含まれる。また、プロセッサ20は、データが、受信したときに、配置されるべきところで、BTEデータ構造においてメモリ・アドレスを含む。BTE論理回路88が進行してパケットを送るときには、バッファ位置のメモリ・アドレスは、レジスタ・ファイル(図示省略)に書込まれる、要求トランザクション論理回路100(図9)であり、レジスタ・ファイルへのポインタとしてTSNを用いている。応答(HDCメッセージ・パケットの形である—図7)がCPU12によって受信されたときに、要求トランザクション論理回路100は、入力メッセージ・パケットに含まれるデータがメモリ28において配置されるバッファの対応メモリ・アドレスに対するレジスタ・ファイル(図示省略)へのポインタとしてパケットのヘッダからのトランザクション・シーケンス番号(TSN)を用いる。

【0125】BTEレジスタ300、302の優先順位(prioritization)を理解するために、CPU12Aから外部装置へのデータの前述の転送は、情報の大きなブロックのものであると想定する。従って、多数のデータ構造は、プロセッサ20によってメモリ28にセットアップされ、(最後を除き)それぞれは、追加データ構造へのチェーン・ポインタ、送られるべきデータを(データ構造304のデータ・バッファ部分304bに)備える総計を含んでいる。優先順位の高い要求は、プロセッサ20によってなされるのが望ましいということをここで想定する。そのような場合には、そのような優先順位の高い要求に対する関連データ構造304は、上述したのと同じ形で、メモリ28に書込まれる。次に、優先順位の高いBTEレジスタ300は、データ構造を位置決めするために必要なBTEアドレス、及びクリアされた要求終了表示ビット(rc0)で書込まれる。しかしならば、BTEレジスタ300を書込むことによって示されたBTE要求は、すぐにはスタートしない。それは、BTEレジスタ302の内容によって始動されたBTE動作がパケット間で休止するまで待つ。更に、BTEレジスタ302の内容により信号が送られたBTE動作は、

BTEレジスタ300の内容によって示されたBTE動作のために中断される。そのBTE動作は、終了するまで進行し、そのときに、BTEレジスタ302の内容によって信号が送られたBTE動作は、再開され、かつBTEレジスタ300が別のBTE動作記述子で再び書込まれない限り終了される。

【0126】— メモリ・コントローラ：図4にちょっと戻ると、インターフェイス装置24a, 24bは、一対のメモリ・コントローラ(MC)26a, 26bを介してメモリ28をアクセスする。Mcsは、インターフェイス装置24とメモリ28の間にフェイルーフースト・インターフェイスを供給する。Mcs26は、(ダイナミック・ランダム・アクセス・メモリ(DRAM)論理回路で実施される)メモリ・アレー28をアクセスするために必要な制御論理回路を供給する。Mcsは、インターフェイス装置24からメモリ要求を受信し、かつ読取り及び書込みを実行すると共にリフレッシュ信号を28でメモリ・アレーを実施するDRAMsに供給する。二つのMcs26a, 26bは、メモリ・アレー28とインターフェイス装置24a, 24bとの間に72ビット・データ経路を供給すべく並列に走り、SBC-DBD-SbD-ECCスキームを用いており、ここでb=4であり、合計100ビット(64データ・ビット+28アドレス・ビット+8チェック・ビット)のうち、72ビット(64データ及び8チェック・ビット)だけが実際にメモリ28に書込まれる。

【0127】図24を参照すると、メモリ28から144ビットのデータをフェッチすべく並列に動作する二つのMcs26a, 26bが示されている。一つのMC(例えば、26a)は、MCとメモリ28の間に72ビット経路330aを形成すべく8チェック・ビットと一緒に連続偶数アドレスで二つの32ビット語を同時にアクセスすべく接続される。他のMC(例えば、26b)は、第2の72ビット経路330bを形成すべく二つの32ビット奇数語を別の8チェック・ビットと一緒に同様にアクセスすべく接続される。この構成は、二つのMcs26a, 26bを一緒に作動させ、かつ最小待ち時間でインターフェイス装置24に64ビット語を同時に供給させ、半分(D0)は、MC26aから生じ、他の半分(D1)は、他のMC26bから生ずる。インターフェイス装置24は、ECCチェック・ビットを生成し、かつ検査する。用いられるECCスキームは、(単一ビット)データ誤りを検出し、かつ修正するだけでなく、全ての二重ビット誤り及び単一DRAMからの4ビットまでの誤りも検出する。フェイルーフースト設計は、インターフェイス24とMCバス25の間、並びに内部レジスタ間のアドレス転送上のパリティを検査する。

【0128】インターフェイス装置24の視点から、メモリ28は、二つの命令でアクセスされる：“N二重語を読取る”及び“N二重語を書込む”。これら指令の両

方は、第1の36ビット転送上のアドレス及び制御と、第2の32ビット転送上のバイト・カウンタとを有するMcs26に入力する。書込みで、Mcs26は、指令を、二重語書込み、または二重語書込みのブロックのいずれかに分解する。読取りで、要求データは、単一二重語読取りまたはブロック読取りフォーマットのいずれかに戻される。“データ有効”と呼ばれる信号は、読取りデータが戻されるかまたは戻されない2サイクル先の時間をインターフェイス装置24に告げる。上記したように、保守プロセッサ(MP18;図1)は、CPUs12へのアクセスの二つの手段を有する。一つは、TNet構造を用いることによって、パケット化された情報を送る(または受信する)ために、ルータ14を含んでいる。より限定されたにもかかわらず、別のものは、システム10の種々の素子(例えば、ルータ14、CPUs12、I/Oパケット・インターフェイス16)に組込まれたOn Line Access Port(オンライン・アクセス・ポート)(OLAP)を通してである。アクセスのこの後者の形は、メモリ・コントローラ26のそれぞれを通してMP18に対する読取り及び書込みアクセスの両方を供給しているOLAP直列ポート285を示す図25に示されている。(図25に示したのは、メモリ・コントローラ26aへのOLAPアクセスである；メモリ・コントローラ26bは、実質的に同一デザインのものである。)ブート・タイムでは、MP18は、プロセッサ20に、それら(プロセッサ20)に動作を開始させ、メモリに一連の命令のイメージを組込ませる命令で、OLAP285に含まれるレジスタを書込み、例えばブート処理を終了する外部(記憶)装置から命令及びデータを転送すべくI/Oに行く。

【0129】また、OLAP285は、誤り表示をMP18に伝達するためにプロセッサ20によって用いられる。例えば、インターフェイス装置24の一つが、メモリ・コントローラ26から受信したデータにおいてパリティ誤りを検出したならば、それは、動作を一時停止する誤り信号を発行すると共に、誤りをMP18に知らせるためにビット位置をOLAP285にセットさせる。メモリ・コントローラ26によって実行される誤り検査(例えば、パリティは、レジスタ読取り動作上の不良を検出する)は、動作を同様に一時停止しかつ誤りが発生したことをOLAP285を介してMP18に知らせる。システムのMP18及び種々のOLAPs(例えば、MC26aのOLAP285)は、IEEE標準1149.1に準じて構成される直列バス287を通して通信する。メモリ・コントローラのアーキテクチャは、Mcs26を実施することにおいて用いる種々の状態マシンを監視することによる誤り検査の特定の形を除き、一般に通常の設計のものである。図26が示すように、MC26aの機能(MC26bについても同様)は、それぞれが複写されかつ比較される、3つの主要な状態マシ

ンによって制御される。マスタ状態マシン・ペア390は、メモリ28にデータを伝達するためにMCADバス25からDRAMデータ・バスにデータ及び命令を得るような、MC26a自身の機能を制御すべく動作する。次に、マスタ状態マシン・ペア390は、MC26aと対応インターフェイス装置24aとの間のバス25上のデータ及びアドレス転送を処理するメモリ制御アドレス/データ(MCAD)状態マシン392にわたり制御を実施する。DRAMデータ・バス上のアドレッシング及びデータ転送、並びに必要なリフレッシュ信号の生成及びシーケンシングは、DRAM状態マシン・ペア394によって制御される。状態マシン・ペア390、392、及び394によって入力されたデジタル状態は、比較回路395によって互いに比較される。比較ミス(miscompare)は、CPU12の動作を一時停止すべく比較ミスを検出している比較回路395からのERROR信号のアサーションを結果として生ずる。

【0130】パケット・ルーティング: 処理システム10の種々の素子間(例えば、CPU s 12A、12BとI/Oパケット・インターフェイス16に結合された装置)で伝達されるメッセージ・パケットは、パケットに含まれる情報(即ち、情報の他のものは、ソース・フィールドとしても、用いることができるが、ヘッダの宛先フィールド、図5(b))により、ルータ14によって“送られる(routed)”。しかしながら、ルータ14の構成及び設計を説明する前に、CPU s 12とルータ14との間、またはルータ14とI/Oパケット・インターフェイス16との間のTNetリンクL上でメッセージを伝達するために用いるプロトコルをまず理解することは、有利である。まず、各TNetリンクLは、受信及び送信能力(機能)の両方を有するポートを介して処理システム10の素子(例えば、ルータ14A)に接続する。素子の各送信ポートは、記号毎の、メッセージ・パケットの同期送信に用いられる送信クロック(T_Clk)信号を供給しなければならない。記号は、送信の受信端のクロック同期FIFOが同期を維持するようにT_Clkの各及び全てのクロック・サイクル(即ち、各クロック周期)で送信される。

【0131】クロック同期は、処理システム10が動作されるモードに依存する。CPU s 12A及び12Bが、例えば、互いに独立(個別)に動作するシンプレックス・モードで動作しているならば、ルータ14とCPU s 12との間のクロッキングは、“近周波数”である;即ち、CPU s 12及びCPU sに直接接続するルータ14によって用いられるクロックは、互いに関してドリフトしうる。逆に、処理システム10がデュプレックス・モードで動作する(例えば、CPU sが同期された、ロックステップ動作で動作する)ときには、それらが接続するルータ14とCPU s 12との間のクロッキングは、“周波数封じ込み(位相封じ込みである必要

はない)である。処理システム10の種々の素子間のデータ・パケットのフローは、いつでも、パケット内さえも、現れうる、指令記号によって制御される。(表1を参照して)上記で考慮したように、指令記号は、全て0である上位ビットによって識別される。それらの指令記号は、次の通りである。

【0132】IDLE: IDLE指令記号は、送るべき他の指令記号またはデータ・パケットが存在しないときにクロック毎に送信される。IDLE指令記号は、TNetリンク上でパケットまたは指令記号間の空白詰め物(space filler)として作用する。

BUSY: BUSY指令記号は、受信装置がデータ記号を受容することができなくなるときに送られる。

FILL: FILL指令記号は、それが記号を送っている受信素子が使用中であるということを送信素子が知る(例えば、BUSY指令記号の受信により)ときに送信素子によってメッセージ・パケットに注入される。

HALT: この指令記号は、CPU12またはMP18によって始動(起動)され、かつ全てのCPU s 12及びある一定のI/O装置によるソフトウェア作用を必要とする事象を伝達すべくルータ14により処理システム10の全ての素子に発布される(広められる)。HALT指令記号は、I/Oアクティビティを始動(起動)することを停止することが必要であることをシステム10の全てのCPU s 12に素早く通知する機構を供給する。

【0133】OTHER LINK BAD (OLB): CPU12に接続されかつデュプレックス・モードで動作しているルータ14がCPU s 12の一つから受信する指令記号またはパケットに誤りを検出し、かつCPU s 12の他のものから受信する指令記号またはパケットに誤りを検出しないときに、ルータ14は、よい(良好な)パケットまたは指令記号を送付したCPU12にOLB指令記号を送る。また、この指令記号は、デュプレックス・モードにおいてのみ、CRC誤り、指令記号誤り、及びプロトコル違反誤りに応じて送られる。OLB及びTLB(以下に説明する)指令記号は、デュプレックスされたCPU s 12へ同時に送られる;即ち、TLB指令記号は、そこから誤ったパケットまたは記号が受信され、または誤りに気づき(error noted)、かつそこから実質的に同時にOLB記号がデュプレックスされたペアの他のCPU12に送られる、ようなCPU12に送られる。

READY: この指令記号は、先に使用中の素子がいま追加データを受容できるときに送られる。

SKIP: この指令記号は、随意にスキップされうるクロック・サイクルを示す。この指令記号は、(1)各記号を転送し、かつそれを各受信クロック同期FIFOに装填し、かつ(2)FIFOからの記号を検索する、

二つのクロック信号間で同期を維持することへの補助として近周波数動作に関連して用いられる。

【0134】SLEEP：この指令記号は、READY指令記号（以下に説明する）が受信されるまで追加の packets（もしあれば、現在送信されているものの後）が特定のリンクLにわたり送られえないことを示すべく処理システム10のあらゆる素子によって送られる。

SOFT RESET (SRST)：SRST指令記号は、CPU s 12とルータ14A、14Bとの間の記号転送を同期し、続いて、デュプレックス動作に対して同一の状態にCPU s 12を配置するために用いられる処理（以下に説明する、“同期”及び“再統一(reintegration)”）の間中にトリガとして用いられる。

SYNC：SYNC指令記号は、デュプレックス・モードに入る前に、または以下に完全に説明するように、デュプレックス・モードにおいて同期を要求するときに、CPU s 12とルータ14A、14Bとの間に周波数封じ込み同期を確立すべく処理システム10（即ち、サブプロセッサシステム10A/10B）のCPU 12へルータ14によって送られる。SYNC指令記号は、例えば、Synchronization及びReintegrationのセクションで以下に更に説明するように、システム・オペレーティング・モードを切り替える（即ち、シンプレックスからデュプレックスまたはデュプレックスからシンプレックス）ためにSRST指令記号に関連して用いられる。

【0135】THIS LINK BAD (TLB)：

TNetリンクLから記号を受信しているシステム素子（例えば、ルータ、CPU、またはI/O装置）が指令記号またはパケットを受信しているときに誤りに気付くときには、それは、ファシリティ・パケットまたは記号を送付したシステム素子にTLB指令記号を送り返す。それは、CRC誤り、指令記号誤り、またはプロトコル違反誤りに応じて通常送られる。

I OWN YOU (IOY)：IOY指令記号は、送信CPUからのデータを選択することをルータ14に強要するためにルータ14へCPU12によってのみ

（かつデュプレックス・モードで動作しているときにのみ）送られて、送信CPU12に、実質的に、所有権を与える；非送信CPUからの更なるデータ送信は、無視される。IOY指令記号の実際のビット構造は、Other Link Bad (OLB) 指令記号に用いられるものと同じである。記号のソースがどれかを決定する。IOY/OLB記号がCPU12によって送られたならば、それは、IOY記号として解釈される；IOY/OLB記号がルータによって送られたならば、それは、OLBとして解釈される。換言すると、CPU s 12とルータ14A、14Bとの間で、CPU s だけがIOY指令記号を送りかつルータだけがOLB指令記号を送る。

【0136】DIVERGE (DVRG)：DVRG記号は、デュプレックス動作におけるときに、CPU s から受信したデータ・ストリームの発散が検出されたことをデュプレックスされたCPU s に知らせるために、ルータによって送られる；即ち、ルータは、クロック同期FIFO s から引かれたときに互いに比較される記号の同一ペアを二つのCPU s 12から受信している。DVRG指令記号は、比較ミスに気付いたことをCPU 12に知らせる。CPU s によって受信されたときに、発散検出処理が入力され、それによりどのCPUが故障しているかまたは誤っているか、かつそのCPUの更なる動作を終了することの決定がCPU s によってなされる。

THIS PACKET GOOD (TPG)：パケットの送信者がパケットのCRCが良好であると決定したことを示している、メッセージ・パケットに続く指令記号。より詳細は、以下の“Packet Status”を参照のこと。

THIS PACKET BAD (TPB)：TPB指令記号は、受信素子が受信メッセージ・パケットのCRCが正しくないということを決定したときにTPG指令記号を置換する。

【0137】— Flow Control：ルータ14は、記憶容量が限られており、従って、メッセージ・パケットをルーティングするときには、“蓄積交換(store and forward)”方法の型を用いない；それよりも、それらは、“虫の穴(worm-hole)”ルーティングとして知られるものを実施する：メッセージ・パケットのヘッダは、そのテールが受信される前にルータを通過してそれから出る。これは、上述したBUSY/FILL/READY指令記号を主に用いて、上述した指令記号が処理システム10の種々の素子間（例えば、CPU s 12、ルータ14、等）のメッセージ・フローを制御すべく動作する一つの理由である。このフロー制御は、“バックプレッシャ”と称される。特定のシステム素子は、その受信キュー（例えば、エラスティック（伸縮性）・バッファ506 — 図27）がほとんどいっぱい（充満）であることを決定するときはいつでも、それは、その上でそれが入力メッセージ・パケットを受信しかつ、更なる送信を防ぐことを送信素子に告げるべく関連送信ポートからのBUSY指令記号を、TNetリンクLの他端の送信素子に、送信する、TNetリンクLの双方向能力（機能）を利用する。BUSY指令記号の使用は、“バックプレッシャ”をアサートすると、ここで称される。CPU s 12またはI/Oパケット・インターフェイス16は、その送信ポートの一つがバックプレッシャされるので“エンド・ノード(end node)”（例えば、CPU12またはI/O装置17 — 図1～図3）がバックプレッシャをアサートしえないが、そのような内部資源が特定のTNetポートにアサートされた

バックプレッシャに関係なく利用可能になるときののみ、内部資源が利用可能になるのを待っている間にそのようなバックプレッシャをアサートしうる。この要求事項（必要事項）の監察することの失敗は、送信ポートが送信できないので、そしてまた関連受信機がバックプレッシャをアサートしているの、受信ポートが受信できないところのバックプレッシャ・デッドロック(backpressure deadlocks)を結果として生じうる。それゆえに、ルータ14だけがバックプレッシャを伝播できる；エンド・ノード（CPU s 12, I/Oパケット・インターフェイス16）は、受信バックプレッシャを送信バックプレッシャに変換することを許可されていない。

【0138】ルータ14は、そのポートに到着している更なるデータ記号がバッファされるか前進されることができないときはいつでも、その受信ポートのいずれか一つにバックプレッシャをアサートしうる。不適当にアドレスされたパケットは、ルータ14によって廃棄される。処理システム10のシステム素子はその上でそれがメッセージ・パケットを送信しているTNetリンクL上のBUSY指令記号を受信したときには、素子は、パケットを送ることを停止し、かつREADY指令記号が送信クロックT_Clkの各クロック・サイクルで受信されるまで、その代わりにFILL指令記号を送ることを始める。FILL指令記号は、送られ続ける。また、関連送信ポートがパケットを送信しない間にBUSY指令記号がTNetリンクL上で受信されるならば、BUSY記号を受信している素子は、それがそのリンク上でREADY記号を続いて受信するまで新しいパケット送信を始動することを慎む。送信ポートは、他の指令記号（READY, BUSY等）を送信する能力（機能）をさもなくば保持する。処理システム10の素子のTNetポートがREADY指令記号を検出するときはいつでも、それは、関連送信ポートでのFILL指令記号の送信を終了し、かつ先の受信BUSY指令記号によって停止されたパケットを送ることを再開するか、またはそれは、IDLE指令記号を注入することを終了しかつ保留しているパケットを送ることを始動するか、またはパケットが利用可能であるまでIDLE指令記号を送り続ける。

【0139】しかしながら、BUSY/READYフロー制御は、他の指令記号の送信に適用されないということが理解されるべきである。上述したように、送信クロック、T_Clk、の全てのサイクルは、指令またはデータ記号の送信を伴うことを思い出す。それゆえに、全てのTNetインターフェイスは、TNetインターフェイスが受信する、関連送信クロック、T_Clk、のクロック・サイクルで新しい指令またはデータ記号を受容する準備ができていなければならない。分かるように、送信された記号を受信するためにTNetリンクLに接続する処理システム10の全ての素子（例えば、ル

ータ14、CPU s 12）は、クロック同期（CS）FIFOを介してそれらの記号を受信する。例えば、上述したように、CPU s 12のインターフェイス装置24は、全てのCS FIFOs 102x, 102y（図10に示される）を含む。各CS FIFO102は、対応TNetリンクLから指令またはデータ記号を受信すべく接続される。CS FIFOは、スピード・マッチング（速度整合）を許容すべく十分な深さを供給しなければならないし、かつエラスティックFIFOsは、メッセージ・パケットの受信の間中のBUSY指令記号の送信と、FILLまたはIDLE指令記号のためによる入力メッセージ・パケットの中断との間で発生しうる遅延を処理するのに十分な深さを供給しなければならない。

また、ルータ14のエラスティックFIFOs 506

（図27）は、送信経路におけるBUSY及びREADY指令記号の注入を許容すべく十分な深さを供給すべきである。例えば、図1～図3を参照すると、CPU12Aが、ルータ14Aのポート3を介してI/Oパケット・インターフェイス16Aの一つによる受信に対してメッセージ・パケットを送信していることを想定する。同時に、また、CPU12Aによって送られるメッセージ・パケットを受信しているその同じI/Oパケット・インターフェイス16Aは、ルータ14Aのポート3に同じ（双方向）TNetリンクL上のメッセージ・パケットを送っている。ルータ14AがI/Oパケット・インターフェイス16Aによって送られるメッセージ・パケットの宛先からのホールドアップ（バックプレッシャ）を経験することを更に想定する。ある時間後、エラスティックFIFO518（図27）は、メッセージ・パケットの送信を一時的に停止すべくI/Oパケット・インターフェイスにリクエストすることをルータ14Aに要求する点まで充たす。従って、ルータ14Aは、ポート3（I/Oパケット・インターフェイス13Aからのメッセージ・トラフィックを受信しているのと同じポート）からBUSY記号を送信する。そのBUSY記号は、CPU12Aからルータ14Aを通して送られるメッセージ・パケットの記号ストリームに挿入される。入力メッセージ・パケットのストリームへのBUSY記号の挿入は、入力パケットの一つの余分な記号を記憶することをルータ14Aに要求する。BUSY記号が送られた後、ルータ14Aは、それがI/Oパケット・インターフェイス16Aからのメッセージ・パケットの切断された送信の受信を再び始められるような時までCPU12Aからの入力メッセージ・パケットの送信を再開することができる。I/Oパケット・インターフェイス16Aに対して割込まれたメッセージ・パケットの再送信を始めるために、ルータ14Aは、ポート3から送られる記号ストリームにREADY信号を挿入し、CPU12Aからのメッセージ・パケットの別の記号を記憶することをルータに再び要求する。

【0140】BUSY/READY指令記号のこのペアは、I/Oパケット・インターフェイス16とCPU12との間の経路に各ルータ14及びCPU12によって挿入することができる。I/Oパケット・インターフェイス16に直接接続されたルータ14は、2n指令記号（n＝経路におけるルータの数＋1）を単一パケットに注入できる。これは、一方向にアサートされたバックプレッシャが、反対方向にアサートされるためにバックプレッシャを必要としないことを確実にするために2nバイトのFIFOが最低レベル・ルータ14（即ち、I/Oパケット・インターフェイス16に最も近いルータ）に要求されるということを意味する。例えば、I/Oパケット・インターフェイス16がルータ14にパケットAを送信し、同時にその同じルータからパケットBを受信し、パケットAを受信しているそのルータがバックプレッシャによりそれを先に進めることができないものと想定する。そのルータは、パケットAを送ることを停止すべくI/Oパケット・インターフェイス16に知らせるためにBUSY信号をパケットBに注入しなければならない。パケットBに注入されたBUSY指令記号は、一つだけFIFOの深さを増加する一つのデータ記号を変位する。READYを注入することによるバックプレッシャの後続の除去は、パケットBにおける別のデータ・バイトを変位する。パケットAが次のルータに進むと、処理は、繰り返される。ルータ14がFIFOが処理できるよりも多くのデータ・バイトを変位したならば、それは、パケットBのソースにバックプレッシャをアサートしなければならない。

【0141】－ パケット状態：各送信されたパケットは、TPGまたはTPB指令記号がすぐ後に続き、関連パケットのインテグリティを報告する。パケットがそこで始まるシステム素子は、適切なTPGまたはTPB指令記号を挿入する。ルータ14は、付随しているCRCを確認し、かつそれらがソース（例えば、I/Oパケット・インターフェイス16またはCPU12）から宛先（例えば、CPU12AまたはI/Oパケット・インターフェイス16）にフローするときに全てのパケットに対して種々のプロトコル検査を実行する。問題のフローの経路における、ルータ14が入力パケット上に誤りを検出し、かつパケットがTPG指令記号で終結する（パケットが良好であることを示している）ならば、ルータは、TPG指令記号をTPB指令記号で置換する。TPB記号へのTPG指令記号の変更をもたらすことができる誤りは、検査したときにCRCデータによる受信データを確かめることの失敗に加えて、使用したプロトコルによって許容されるものよりも大きい長さを有するパケットを含む。あらゆるパケット長を用いることができるが、ここでは、パケットは、状態（TPG/TPB）記号を含んで、1024記号に制限される。受信したパケットが、この制限よりも多くを有しているとして検出さ

れたならば、受信ルータは、1024番目の記号におけるTPB指令記号でパケットを終了し、パケットの残りを廃棄する。パケット長のこの制限は、それを絶え間なくバブルさせ、かつTNetネットワークを詰まらせるパケット送信素子において発生することからフォルトを排除する誤り検査技術である。

【0142】TPB指令記号に続いてパケットを受信するルータ14は、それ自身の結果にかかわらず、不変更のTPB指令記号を先に進める。

【0143】－ SLEEP Protocol：SLEEPプロトコルは、以下に説明する、保守インターフェイス（オン・ライン・アクセス・ポート・OLAP）を介して保守プロセッサによって始動される。SLEEPプロトコルは、パケット境界で一つ以上のTNetリンクLを静止する（quiesce）ための機構を供給する。一つのシステム10を再統一するためにモードを変える（例えば、デブプレックスからシンプレックス）ことが必要である。ルータ14は、データ損失または汚染をもたらすことなくモードを変えるためにアイドル（idle）（処理にパケットがない）でなければならない。SLEEP指令記号を受信したときに、処理システム10の受信素子は、そのTNetリンクL上の許可された指令記号だけを送信しなければならない関連送信ポート上の新しいパケットの送信の始動を抑制する。（例外は、再統一を取り扱うセクションで以下に説明する、自己アドレス型AtomicWriteメッセージ・パケットである。）SLEEP指令記号を受信されるときに送信されるパケットは、終了するまで普通に送信される。しかしながら、SLEEP指令記号を受信される受信ポートに関連した送信ポートは、許可された指令記号（例えば、BUSY, READY, IDLE, FILL）を送信し続けるが、READY指令記号がその関連受信ポートで受信されるまで送信のための新しいパケットを始動しない。

【0144】－ HALT Protocol：HALT指令記号は、I/Oアクティビティ（即ち、CPU12とI/Oパケット・インターフェイス16との間のメッセージ送信、または異なるCPU12間のメッセージ送信）を終了することが必要であることを処理システム10の全てのCPU12に素早く知らせる機構を供給する。各ルータ14は、HALT指令記号がCPU12から受信されたときに、受信ルータ14がその送信ポートのそれぞれからHALT指令記号を伝播し、かつそのシステム停止イネーブル・ビットをクリアするようにOLAP285'（図27）を通してMP18によってセットすることができるシステムHALTイネーブル構成レジスタを有する。ルータ14は、システム停止イネーブル・ビットがクリアされた状態であるときに受信する全てのHALT指令記号を無視する。このようにして、システム停止イネーブル・ビットは、停止機能のための

ソフトウェア・セット可能イネーブル、並びに一度第1のHALT指令記号がアサートされるとHALT指令記号の無限サイクリングを防ぐことの両方に機能する。

(インターフェイス装置24)のそれらの受信ポートのいずれかでHALT指令記号を受信するCPU sは、システム停止割込みがイネーブルされる(即ち、マスク・レジスタ282の関連処理(associated disposition)が割込みをイネーブルする;図21)ならば割込みレジスタ280に割込みをポストする。

【0145】CPU s 12は、HALT処理をディスエーブ(禁止)するための機能(能力)を備えうる。それゆえに、例えば、インターフェイス装置24の構成レジスタ75は、所定の状態(例えば、ZERO)にセットされたときにHALT処理をディスエーブするが、誤りとしてHALT記号の検出を知らせる、“停止イネーブル・レジスタ”を含むことができる。

【0146】ルータ・アーキテクチャ:ここで図27を参照すると、ルータ14Aの略ブロック図が示されている。処理システム10の他のルータ14(例えば、ルータ14B, 14'等)は、実質的に同一な構成のものであり、従って、ルータ14Aに関する説明は、他のルータ14に同様に適用する。図27に示すように、ルータ14Aは、それぞれがポート入力502(5020, . . . , 5025)及び出力504(5040, . . . , 5045)を含んでいる、6つのTNetポート0, . . . , 5を含む。各ポート出力504は、上述したそれから出ている10の信号回線を有する:並列9ビット指令/データ記号を送信する9つの信号回線、及び関連送信クロック(T_Clk)を運ぶ信号回線。同様に、ポート入力502のそれぞれは、データ、受信クロック(Rcv_Clk)を含んでいる10の並列信号を受信すべく接続する。更に示すように、各ポート入力502は、入力論理回路505及びそれをクロスバー・スイッチ500に印加する前に入力メッセージ・パケットを受信しかつバッファするエラスティックFIFO506を含む。クロスバー論理回路500は、メッセージ・パケットのDestination(宛先)IDに含まれる情報によりポート出力504へポート入力502によって受信したメッセージ・パケットを送るべく動作する。クロスバー論理回路500は、真のクロスバー・スイッチとして動作して、そのポート出力504がパケットを受信しているポート入力502に関連したとしても(例えば、ポート入力5022及びポート出力5042)、ポート入力502で受信したメッセージ・パケットをポート出力504に送らせる。クロスバー論理回路500は、ポート入力502の対応するものからポート出力504に二つ以上のメッセージ・パケットを送るようにも構成されている。クロスバー論理回路500は、その構成についての更なる説明が必要でないように通常の設計のものである。

【0147】シェーディングにより図において強調された、ルータ14Aのポートの二つ、4及び5は、他のものとはある程度異なって構成される;これら二つのポートは、一対のCPU s 12に(TNetリンクLx及びLyによって)直接接続するそれらのポートとして用いられることを意図している。これらのポート4, 5に対するポート入力5024, 5025は、処理システム10がデュプレックス・モード動作に対してセットされるときに周波数封じ込み環境で動作すべく構成される。更に、デュプレックス・モードであるときに、入力ポート0-5のいずれか一つで受信され、かつルータが接続するCPU s 12のいずれか一つに向けられたメッセージ・パケットは、クロスバー論理回路500によって繰り返え(再現)され、かつそれらが実質的に同時に、記号毎に、同じ記号を接続するCPU sへ送信すべくロックステップ・ファッションで動作する二つのポート出力5044, 5045の両方に送られる。デュプレックス・モードで動作していない(即ち、シンプレックス・モード)ときには、ポート入力5024, 5025、及び全ての他のポート入力は、近周波数モードで動作する。更に、また、ルータ4及び5に対する入力論理回路502は、二つのCPU sから受信した指令/データ記号の記号毎の比較を実行すべく、CPU s 12A, 12Bがデュプレックス・モードであるときに、動作する比較回路を備えている。従って、図28に示すように、ポート入力5024, 5025は、CPU sから指令/データ記号を受信し、クロック同期FIFOs 518(以下に更に説明する)を通してそれらを渡し、かつクロック同期FIFOsを出て行く各記号をゲート型比較回路517と比較する。デュプレックス動作が入力されたときに、制御論理回路509の構成レジスタ(図示省略)は、DPX信号をアサートする状態にセットされる。DPX信号は、ポート4及び5に対するルータ入力論理回路502の二つの同期FIFOs 518から出て行く記号の記号毎の比較をアクティベート(活性化)すべく制御論理回路509からゲート型比較回路517へ伝達される。もちろん、DPXビットが制御論理回路509にセットされなかったときには、比較がディスエーブされる。

【0148】同一記号ストリームである、デュプレックスされたCPU s 12からのメッセージ・トラフィックは、ポート入力5024, 5025によって受信され、一つのポート入力によって受信されたストリームの各記号は、他のポート入力によって、実質的に同時に、受信されたものと同一である。デュプレックス・モードにおいて同期を維持するために、CPU s 12に送信するルータ14Aの二つのポート出力は、ロックステップで動作しなければならない;即ち、ポート出力は、サイクル間に応じて(on a cycle-to-cycle basis)同じ記号がCPU s 12の両方に送られなければならないように動

作しなければならない。それゆえに、図4を参照すると、ルータ14Aのポート0-5（図27）の一つで受信され、かつCPU s 12に向けられた記号ストリームは、同一記号が実質的に同時にCPU s によって受信されるように、デュプレックス動作で、両方のCPU s 12に進められ（送られ）なければならない。（CPU s 12は、デュプレックス・モードであるときには、ルータ14によって複写され、かつ両方のCPU s に戻される、自己アドレス型メッセージ・パケットを送ることができる。）CPU s 12に直接結合される出力論理回路装置504₄、504₅は、両方ともに同期したファッションで（メッセージ・パケットのDestinationフィールドがデュプレックスされたCPU s 12の一つ、例えば、CPU 12Aだけを識別しても）クロスバ論理回路500から記号を受信し、二つのCPU s 12へ実質的に同期ファッションでそれらの記号を与える。もちろん、CPU s 12（より正確には、関連インターフェイス装置24）は、図11に示したものと実質的に同じ構成の同期FIFOsで送信された記号を受信して、それを伴って記号がCPU s 12によって受信される多少の実時間位相差が存在していても、二つのCPU s 12間で維持されるクロッキングは、同じ記号が同じ命令サイクルで両方のCPU s 12によってFIFO構造から引き出されることを確実にして、デュプレックス動作モードによって要求されるCPU s 12の同期した、ロックステップ動作を維持する。

【0149】ポート入力502のより詳細な図（図29及び図31）の説明に関して見られるように、ルーティング制御は、（オン・ライン・アクセス・ポート285'及び直列バッファ19Aを介して；図1参照）保守プロセッサ18により制御論理回路509に含まれるレジスタに書込まれた構成データと共に、ポート入力502の論理回路によって主に行われる。ルータ14Aは、適切な動作を確実にするためにルータ14Aを構成する種々のコンポーネントの検査を実行する自己検査論理回路511を更に含む。一般に、自己検査論理回路511は、内部パリティ検査、状態マシンの違法状態検出、及び複写された論理回路の出力の比較のような動作を実行する。実行される自己検出は、通常の特質のものである。ルータ14Aの同期動作は、クロック論理回路510によって生成される（局所）検査信号により実行される。ルータ14の各出力ポート504は、TNetリンクL上の記号を伝達するために、上述した、フロー制御プロトコルの要求事項を実施すべく構成される。また、各ポート入力502の入力論理回路505は、受信したSKIP指令記号を除去することによって — 少なくとも近周波数環境で記号を送っているポートに対して — 同期を維持することを補佐する。SKIP指令記号は、このコンテキストにおいて、実質的に、クロック・サイクルをスキップさせて低速な受信機に高速な受信機

からデータを受容させる、位置-保持記号として用いられる。TNetリンクLの端末における装置が異なるクロックで動作するので、近周波数環境で動作しているときに、一つのクロックが多少の量だけ他のものよりも速いということは、相対的に確実である。検査されないままであったならば、高速-送信素子から記号を受信している低速-受信素子は、低速-受信素子の入力クロック同期FIFOに負荷を掛け過ぎる。即ち、高速クロックによりそこに置かれた低速クロックがクロック同期FIFOから記号を引く（pull）ために用いられたならば、結果として検査同期FIFOは、オーバーフローするであろう。

【0150】ここに採り入れられた好ましい技術は、同期化FIFOから記号を引くために用いる局所クロックよりも周波数が多少高いFIFOに記号を押す（push）T_{CLK}信号によりルータ14（またはCPU 12）のクロック同期FIFO（即ち、クロック同期FIFO 518；図29）のオーバーフローの可能性を回避、または少なくとも最小にすべく記号ストリームにSKIP記号を周期的に挿入することである。（FIFOへの）押し（push）動作をバイパスするためにSKIP記号を用いることは、SKIP指令記号が受信される毎にFIFOの押しポインタを失速させる効果を有するので、クロック同期FIFOに関する限り、SKIP記号を伴った送信クロックが失われていた（欠けていた）。それゆえに、ポート入力502のそれぞれにおける論理回路は、何もFIFOに押されないが、記号が引かれるように、近周波数クロッキング環境における同期に対してSKIP指令記号を認識し、かつその受信をキー・オフ（key off）する。SKIP記号は、おおよそ512送信機クロック毎に挿入されるのが好ましい。50Mhz速度で記号がリンクLに送信される（例えば、CPU 12とルータ14との間、またはルータ14間、またはルータ14とI/Oインターフェイス装置16Aとの間 — 図1〜図3）と仮定すれば、これは、最悪の場合、2000ppmの周波数差を許容する。

【0151】各ポート入力502のエラスティックFIFOs 506は、通常設計のものであり、例えば、通過中にメッセージ・パケットの中にフロー制御及び指令記号を挿入することによってもたらされた、記号ストリームにおけるジッタを吸収しかつ平滑にすることを補助するために用いられる。たぶん、最も重要なことは、エラスティックFIFOs 506は、出力ポートが使用中のときに入力メッセージ・トラフィックのバッファリングを許容する。システム10の他の素子のように、ルータ14Aは、宛先装置へ受信したメッセージ・パケットを送るときに“バックプレッシャ”を経験しうるし、かつ宛先装置は、更なる記号を受信できないことを瞬間的に知らせる（例えば、BUSY指令記号）。バックプレッシャの適切な実施は、エラスティックFIFOs 506

が、先の装置（例えば、ルータにメッセージ・パケットを供給する装置）が（受信されかつクロック同期FIFOsに押されるが、エラスティックFIFOsに渡されない）FILLまたはIDLE記号を供給することによりBUSY記号に回答できるまで宛先装置が受信を停止した後で入力記号を受信しかつ保持するのに十分に大きな深さ有する（即ち、十分な数の記憶位置を有する）ことを必要とする。要約すると、各エラスティックFIFO506は、送信装置が送くることを一時的に停止できるまで記号を記憶し続けるために十分な空間をもたなければならない。

【0152】記号ストリームにおけるジッタを低減することを補助するために、エラスティックFIFOs506は、高及び低“水位標(water marks)”で動作する。エラスティックFIFO506が充滿し始め、かつ高水位標に達したならば、バックプレッシャ記号（例えば、BUSY）は、記号ストリームを受信している受信ポートに対応している送信ポートから送信される。例えば、記号ストリームがルータ・ポート入力502₃によって受信されており、かつエラスティックFIFO506₃を制御するために用いられるFIFO制御論理回路546がFIFOが充滿している（即ち、高水位標を通過した）ことを示すならば、入力ポート502₃は、BUSY記号を送信させるべく対応出力ポート504₃に知らせる。BUSY状態は、ポート出力504₃がREADY記号を送るべく知らされて、記号ストリームのフローの再開を要求するときに、FIFO制御論理回路546（図29）によって決定されたように、エラスティックFIFO506₃の深さが低水位標以下であるまでルータ14（及びパケットを送っていたTNetリンクLの他端における装置）によって維持される。TNetリンクLの他端において、メッセージ・パケットを送っていた装置は、関連出力指令リンク上に送信されたFILL指令記号で入力リンクでのBUSY指令記号の受信に回答する。送信装置は、BUSY指令記号を送った装置がREADY記号を送るまで、メッセージ・パケットの更なる送信を手控えて（見合わせて）、FILL記号を送り続ける。メッセージ・パケットの送信は、終了まで、またはバックプレッシャが受信機によって再びアサートされるまで、再開する。

【0153】エラスティックFIFOs506がこの“バックプレッシャ”ジッタを処理するために十分に大きくなければならないばかりでなく、それは、制御記号が他の方向におけるTNetリンクLの制御に対して記号ストリームに挿入されている間にFIFOに累積するデータ記号を記憶することもできなければならないということに注目すべきである。BUSY/READY組合せは、ポート出力504から2サイクルを奪ってそのポート出力504を供給しているエラスティックFIFO506を2文字(two characters)でいっぱいにする。ジ

ッタを最小に保つために、エラスティックFIFOs506のサイジング（及び高及び低水位標の配置）は、バックプレッシャがアサートされる前にストリームの中に挿入されることを少なくとも二つの文字、好ましくはそれ以上に対して許容しなければならない。ここに記載されたシステム的环境内で、エラスティックFIFOs506は、96記号を一時的に記憶することが可能である。ルータ14Aは、バックプレッシャが要求される前に挿入されることを所定数の記号に対して許容する（バックプレッシャは、所定数が受信されかつ一時的に記憶された後で次の記号に発行される）。エラスティックFIFOs506の96記号深さは、所定数の記号の標準蓄積(buildup)、及びポート入力502がデータを受容することをやめる（バックプレッシャをアサートする）か、またはオーバーフローによるデータの損失の醜行(ignominy)に見舞われなければならない前にバックプレッシャ遅延の12サイクルを許容する。

【0154】ポート入力502のそれぞれは、一つの説明が全てに適用されるように実質的に同一に構成される。従って、図29に示すように、ポート0に対するポート入力502₀の詳細ブロック図が示される。ポート入力502₀は、それが付随する送信クロック(T_{clk})によって一時的に記憶される入力レジスタ516で各9ビット・データ/指令記号を受信する。次に、受信した記号は、またT_{clk}によって、入力レジスタ516から伝達されかつクロック同期化FIFO518に印加される。クロック同期FIFO518は、CPU12のインターフェイス装置24に用いられる、図8A及び図8Bに示したものと論理的に同じである。ここで、図29が示すように、クロック同期FIFO518は、入力レジスタ516の出力を、並列に、受信する複数のレジスタ520を含んでいる。レジスタ520のそれぞれと関連するのは、図30に詳細に示され、かつ以下に説明される、2段階妥当性(V)ビット・シンクロナイザ522である。レジスタ20のそれぞれの内容は、各関連2段階妥当性ビット・シンクロナイザ522の1ビット内容と一緒に、マルチプレクサ524に印加され、かつ選択されたレジスタ/シンクロナイザがFIFOから引かれ、かつ一對のレジスタ526によりエラスティックFIFO506に結合される。入力レジスタ516の内容を受信するレジスタ520の選択は、押しポイント論理回路装置530によって供給されたPush Select信号の状態によって決定される；そして、レジスタ526に、MUX524を介して、その内容を供給するレジスタ520の選択は、引きポイント論理回路532によって供給されたPull Select信号の状態によって決定される。押し及び引きポイント論理回路530、532は、syncFIFO制御論理回路534の制御下である。syncFIFO制御論理回路534は、押しポイント論理回路530（並びに

入力レジスタ516)を動作しかつ押しポインタ論理回路530によって選択されたレジスタ520を装填すべく入力T_Clkを受信する。同様に、同期FIFO制御論理回路534は、引きポインタ論理回路532を制御すべくルータ(Rcv_Clk)に対して局所的であるクロック信号を受信する。

【0155】ちょっと脇道にそれて、図30を参照すると、イネーブル機能を有するD型フリップフロップ541、遅延素子541a、ORゲート541b、(以下に示す真理値表に示される機能を提供するためのセット/リセット/イネーブル能力を有する)D型フリップフロップ542、及びD型フリップフロップ543を含んでいる妥当性ビット・シンクロナイザ522が詳細に示されている。D型フリップフロップ541は、そのデータ(D)入力でSKIP検査論理回路540の出力を受信すべく結合される。フリップフロップ541のEnable(イネーブル)入力は、押しポインタ530、Push Select(選択)、によって供給される復号を受信し、かつフリップフロップ541のクロック(Clk)は、入力記号を付随している入力送信クロック(T_Clk)を受信する。フリップフロップ541の出力(Q)は、ORゲート541bの一つの入力に印加され、かつ遅延素子541aを通して他の入力にも印加される。フリップフロップ541の出力(Q)は、ポインタ論理回路530(図29)からのPush Select信号が、妥当性ビット・シンクロナイザが次の記

号 — SKIP記号でないならば — の受信に対して関連するFIFOのレジスタ520を選択するとき(論理“1”レベルに)セットされる。

【0156】遅延素子541a及びORゲート541bは、通常設計のバース引伸し回路を形成すべく動作し、フリップフロップ542のSet入力での信号が少なくとも1クロック期間の持続時間を有することを確実にする。その場合でありかつ(ルータに対して)局所的なRcv_Clkという知識及び受信したT_Clk信号が、もし同一でないならば、類似な周波数を有することを想定すると、Rcv_Clkの少なくとも一つのアクティブ・トランザクションが、フリップフロップの出力(Q)をセットすることによってフリップフロップ542に引伸ばされた信号を記録させるということが明らかになる(以下の、真理値表を参照)。D型フリップフロップ543は、同期の追加段階として作用し、局所Rec_Clkに関してV出力で安定レベルを確実にする。Pull Select信号、引きポインタ532の復号は、フリップフロップ542のイネーブル入力に接続し、関連レジスタ520が読取られたときにPull信号(syncFIFO制御装置534からの周期パルス)に妥当性シンクロナイザ522上の妥当性ビットをクリアさせる。

【0157】

【表4】

表 4
真理値表

セット	Rst	イネーブル	O_n	O_{n+1}
1	X	X	X	1
0	X	0	0	0
0	X	0	1	1
0	1	1	X	0
0	0	1	1	1
0	0	1	0	0

【0158】要約すると、妥当性シンクロナイザ522は、有効記号であるとその記号を識別するために記号がFIFO518のレジスタ520に装填されるときに“妥当性”(V)信号をアサートすべく動作する。他方、記号がSKIP記号であったならば、SKIP検査論理回路540の出力は、LOWに行き、フリップフロップ541(即ち、データ(Q)出力)をゼロのままにさせて、関連記号が有効でないことを示し、かつ無視されるべきある。図29を続けると、入力レジスタ516の内容は、SKIP検査論理回路540にも印加される。SKIP指令記号の受信は、SKIP制御論理回路540によって検出されたときには、押しポインタ論理回路530の動作を禁止(抑制)すべく動作し、かつT_Clkの一付随クロック周期に対してクロック同期FIFO518にその記号を装填することを排除する。S

KIP指令記号の受信は、押しポインタ530を先に進めないかまたは妥当性ビットVをセットさせず、実質的に、押しサイドによるSKIP記号の受信のFIFO無知の引きサイドを保持する。レジスタ・パイプライン526から渡された入力データ/指令記号は、入力ストリームの指令記号が復号されかつFIFO制御論理回路546を制御するために用いられる。エラスティックFIFO5060を動作することに加えて、FIFO制御論理回路546は、クロスバール論理回路500を介してポート入力5020から記号を受信するポート出力504に対して必要なハンドシェーク信号を生成すべく動作する。

【0159】指令/データ記号は、リンク・レベル“キープアライブ(keep-alive)”プロトコル(以下に説明する)、メッセージ・パケット終了検査、等を含んでい

る、リンク・レベル及びパケット・プロトコルを確認すべく動作するプロトコル及びパケット検査論理回路550にも印加される。(見出されたときに、記号ストリームから抽出される) 指令記号でない、即ち、データ記号である、それらの記号は、アクセスされたときに、それからクロスバー論理回路500に伝達される、エラスティックFIFO506に渡されかつ記憶される。また、メッセージ・パケットのDestination IDは、ターゲット・ポート選択論理回路560にも伝達される。ターゲット・ポート選択論理回路560は、メッセージが送信のために送られる、ポート出力504の“ターゲット・ポート”アドレスを、受信したDestination ID及びルータの構成レジスタのある情報から決定すべく動作する。ターゲット・ポート選択論理回路560は、適切なクロス接続を行うためにクロスバー論理回路500に印加されかつそれによって用いられる3ビット・コードを発生する。しかしながら、選択ポート出力504は、ポート入力5020からメッセージ・パケットを受信すべく“イネーブル”されなければならない。この目的のために、ポート入力5020は、どのポート出力504がポート入力5020からメッセージ・パケットを受信すべく許可されるかという情報を含んでいる、6ビット・ポート・イネーブル・レジスタ562を含む。ポート・イネーブル・レジスタ562の各ビット位置は、一つのポート出力504に対応しており、かつ特定ビット位置の状態により、対応ポート出力は、ポート入力からそれに送られるメッセージ・トラフィックを有すべく“イネーブル”されうるか、または“ディスエーブル”されて、ポート入力5020からそれに送られるメッセージ・トラフィックを排除する。例えば、ポート入力5020がターゲット・ポート選択論理回路に宛先ポートとしてポート出力5044を識別させる宛先情報を有しているメッセージ・パケットを受信し始めると想定する。しかしながら、ポート・イネーブル・レジスタ562の状態は、ポート出力5044がポート入力5020からメッセージ・トラフィックを受信すべく許可されないようであると更に想定する。この場合には、ポート・イネーブル・レジスタ562の内容は、ターゲット・ポート選択論理回路506によって発生された選択情報がクロスバー論理回路500に印加されることを禁止すべく動作する。それよりも、パケットは、降下され、かつルータ14Aが、パケットが受信されたポートに対して許可されないポート向けのパケットを受信したということを示すべく誤り信号が生成される。誤りは、OLAP285'を介してMP18に知らされる(図27)。

【0160】従って、ポート・イネーブル機能(特徴)は、ルータ14を通るある一定のルーティング経路を選択的に防止すべく動作する。この機能は、デッドロック状態を防止する重要な機構でありうる。デッドロック状

態は、メッセージを伝達すべく用いるネットワークが、ルーティング装置及び相互接続リンクによって形成された“ルーティング・ループ(routing loops)”を含むときに発生する。別のメッセージが既にそのポートから送られる処理中であるので、一つのルーティング装置で受信したメッセージが特定のポートから送られることからブロックされることが発生する。しかしながら、次いで、他のメッセージも第3のメッセージにより別のルーティング装置でブロックされる、等。全てのメッセージは、それぞれ環状ループでブロックされる。ループの各メッセージがループの別のメッセージによってブロックされ、かつループの別のメッセージをブロックしているので、何も移動しない;メッセージは、デッドロックされる。適切な設計なしで、大きなルーティング・ネットワークは、一群のメッセージ・パケットのそれぞれが通信リンクへのアクセスを獲得する前に別のものが進行するのを待たなければならないような環状(循環)依存性(circular dependencies)のために、通信ネットワークを通して更なる前進を行うことができないメッセージ・パケットのグループを結果として生ずるデッドロックに対する多数の環境の可能性を生じさせることができる。ルータを通るある一定の通信経路をディスエーブルすることにより、可能なルーティング・ループを除去することができ、それにより、デッドロックが発生する可能性を除去することができる。

【0161】もちろん、ルーティング・ループに対抗する防御の最初のライン及びデッドロックの可能性は、適切なルーティング情報が、入力メッセージ・パケットがルーティング・ループの部分でありうるルータ14のポートから送られないようにターゲット・ポート・アドレスを選択するために用いられることを確実にすることである。しかし、ポート・イネーブル・レジスタによって達成されるような、ルータ14を通るある一定のルーティング経路をディスエーブルする能力(機能)は、ルーティングまたは他の誤りがデッドロック状態を結果として生じないことを確実にする。この概念の実施は、以下により詳細に説明される。再度、図29を続けると、入力メッセージ・パケットのヘッダが受信されると、Destination IDsは、ターゲット・ポート選択論理回路560に逐次的に渡されかつ先着順にそこで試験される。ターゲット・ポート選択論理回路560は、指定ポート出力を識別する、ターゲット・ポート・アドレスを生ずる。上記したように、選択ポート出力504がポート入力に対してイネーブルされることを条件として、そのアドレスは、メッセージ・パケットを受信するエラスティックFIFO506の出力を適切なポート出力に伝達する適切なクロスバー選択を行うためにクロスバー論理回路500に印加される。(ルーティング14がCPUs12への直接的なTNet接続を有するものであり、かつデュプレックス・モードで動作し

ているならば、CPU s に向かう入力メッセージ・パケットは、メッセージ・パケットを同時に両方のポート出力504₄及び504₅にルーティングすることによりクロスバ理論回路装置によって複写される。)

ターゲット・ポート選択論理回路560は、図31により詳細に示されており、かつ入力パケットの3バイトDestination IDをポート出力502のエラストティックFIFOs 506 (図27~図29) から受信する、宛先レジスタ570を含んで示されている。Destination (宛先) IDは、図5 (b) に関して上述した3つのフィールドを含む: Region (領域) ID, Device (装置) ID, 及び経路選択ビット (P) を含んでいる1ビット・フィールド。Region (領域) IDは、名前が示唆するように、領域による宛先を識別し、Device (装置) IDは、その領域内の特定の装置を表わす。経路選択ビット

(P) は、経路 (XまたはY) のどちらが二つのサブプロセッシング装置をアクセスするために用いられるべきかを識別する。

【0162】ルータ14は、例えば、大規模な並列処理アーキテクチャに対して、大きくて、用途が広いルーティング・ネットワークを構築する能力 (機能) を供給する。ルータは、制御論理回路509に含まれるある一定のルータの構成レジスタにセットされた情報によりネットワークにおけるそれらの位置 (即ち、レベル) によって構成される。これらの構成レジスタは、上部領域IDレジスタ509_a、下部領域ID509_b、HiLoレジスタ509_c、デフォルト・ポート・レジスタ509_d、クロスリンク・ポート・レジスタ509_e、デフォルト・レジスタへの経路509_f、装置ID比較レジスタ509_g、及びサイド・レジスタ509_hとして図31に示される。二つの追加構成レジスタは、装置位置としてかつレジスタ509_j及び509_kをそれぞれ伴って図33に示される。これら種々の構成レジスタの内容は、Device ID及びメッセージ・パケットの付随経路選択ビット (P) と一緒に、メッセージ・パケットがクロスバ理論回路500を通して送られるポート出力504の選択を決定する。ルータのレベルは、部分的に、Destination IDのどの部分がターゲット・ポートの選択に用いられるか、及びアルゴリズム・アドレス選択を用いることができるか否かを決定する。この目的のために、Region IDは、二つの重複10ビット・レベル識別に更に分割される。Region IDの内容の上位10ビットは、上部レベルとして画定され、Region IDの下位10ビットは、下部レベルを指定する。両方のレベル識別は、マルチプレクサ572の二つの10ビット入力に対応するものに印加される。マルチプレクサ572は、ルータのレベル (上部または下部) を識別し、かつ選択した10ビットをアドレスとしてルーティング表584に供給する

HiLoレジスタ509_cの内容に応じて二つの10ビット入力の一つを選択する。

【0163】図29及び図31は、それぞれがそれら自身、個別のターゲット・ポート選択論理回路560、及びルーティング表584を有しているようにポート入力502を示す。しかしながら、空間を最小にするために、単一ルーティング表を全ての6つのポート入力502のターゲット・ポート選択論理回路によって共有することができることは、当業者には明らかであろう。マルチプレクサ572の出力は、通常のラウンド・ロビン・アービトレーション方法を用いて、アービトレーティド・ベシスで (arbitrated basis)、(状態及び制御論理回路509に含まれる) ルーティング表584にそれ自身マルチプレクスされうる。ルーティング表のアクセスの結果は、戻されかつマルチプレクサ586の入力に印加される。簡略化のために、このアービトレーション及びマルチプレキシングは、図31に示さない。また、Region IDの4つの上位ビットは、それらが上部領域IDレジスタ509_aの内容と比較される4ビット比較回路574にも印加される。Region IDの下位10ビットは、それらが下部領域IDレジスタ509_bの内容と比較される比較回路578に結合される。

【0164】例えば、ランダム・アクセス・メモリの形でたりうる、ルーティング表584は、複数の3ビット・ターゲット・ポート識別を記憶すべく動作する。Region IDの一つのまたは他の10ビット部分によってアドレスされたときに、ターゲット・ポート選択情報の3つのビットは、マルチプレクサ586の一つの入力に結合される; マルチプレクサ586の他の入力は、デフォルト・ポート・レジスタ509_dの3ビット内容を受信する。マルチプレクサ586によるルーティング表584の出力の選択は、(ルータが上部レベル・ルータであることを示している) ONEのときに、HiLoレジスタ509_cの内容、または (宛先がこの“低レベル・ルータ”と同じ“高領域 (high region)”であることを示している) 比較回路574による上部Region IDレジスタ509_aの内容とRegion IDの4つのMSBsとの間の成功した比較 (successful compare) のいずれかによって行われる。それら条件のいずれも満足されないならば、マルチプレクサ586は、ターゲット・ポート識別としてデフォルト・ポート・レジスタ509_dの (3ビット) 内容を代わりに選択する。ルーティング表584は、どんなサイズのものでもありうる。しかしながら、当業者に明かなように、ルーティング表584のサイズは、ルータが用いられるシステムのアドレス可能素子の数、及び表に対して利用可能な空間のようなファクター (因子) によって指図される。ターゲット・ポート選択論理回路560は、ルーティング表において空間を確保するために、要求されたときに

表ルックアップ技術の使用、または要求されなかったときにアルゴリズム的ルーティングを組み合わせることにより新規な妥協（コンプロミス）を実施する。この組合せは、入力メッセージ・パケットを6つの利用可能なポートの一つへ渡し、かつそれから送信させ、及び非常に多種多様なルーティング能力を供給する。

【0165】マルチプレクサ586によって選択された3ビット・ターゲット・ポート識別は、マルチプレクサ586の出力とクロスリンク・ポート・レジスタ509_eの3ビット内容との間を選択する更なるマルチプレクサ590の一つの（3ビット）入力に伝達される。二つの値のいずれが選択されるかは、入力メッセージの経路選択ビット（P）の状態によって示されるように最終宛先のサイド（即ち、XまたはY）によって決定される。入力メッセージ・パケットの経路選択ビット（P）は、その出力がマルチプレクサ590によって行われる選択に影響を与えるサイド・レジスタ509_hの内容と比較される。ルータが、メッセージ・パケットが向けられたものと同じサイド（XまたはY）上でないならば、コンパレータ592の出力は、クロスリンク・ポート・レジスタ509_eの内容の選択に影響を与える。これは、ルータを含んでいるXまたはYサイドからメッセージ・パケットの宛先を含んでいる他のサイドにメッセージ・パケット直接的または間接的（即ち、別のルータを通す）に送るそのポート出力504にメッセージ・パケットを送る。マルチプレクサ590によって行われる選択は、その選択入力がANDゲート論理回路596の出力を受信するマルチプレクサ594の入力に印加される。マルチプレクサ594は、マルチプレクサ590及びマルチプレクサ598によって供給されるポート・アドレス間を選択する。次いで、マルチプレクサ598は、アルゴリズム・ルーティング論理回路600の出力とデフォルト・ポート・レジスタ509_dの内容との間を選択する。この選択は、Device ID（構成）レジスタ509_gの内容及び入力メッセージのDevice IDの6ビットの選択部分を受信する選択及び比較回路601によって行われる。特に示されていないのは、アルゴリズム・ルーティング論理回路600（図33）のそれぞれ装置ビット位置及び拡張レジスタ509_j、509_kも、選択及び比較回路601に印加されるということである。装置ビット位置及び拡張レジスタ509_j、509_kに含まれる値は、アルゴリズム・ルーティング技術で用いられないDevice IDの高位ビットだけがDevice IDレジスタ509_gの内容と比較されるようにメッセージのDevice IDビットをマスクすべく動作する。

【0166】メッセージのRegion IDの選択された（マスクされた）ビットとDevice IDレジスタ509_gの内容との間のマッチは、可能なターゲット・アドレスとしてマルチプレクサ598でアルゴリズム

・ルータ600の結果を選択することを結果として生ずる。例えば、Region IDが“abcdef”

（aが高位ビットである）であり、かつ装置ビット位置及び拡張レジスタ509_j、509_kに含まれる値がビット“def”がアルゴリズム処理に用いられるのようなものであるならば、Region IDのビット“abc”は、選択及び比較回路601によってDevice IDレジスタ509_gの内容と比較される。逆に、ビット“def”がアルゴリズム・ルーティングに用いられるならば、ビット“ab”だけがDevice IDレジスタ509_gの内容と比較される。また、メッセージのDevice IDのどのビットがアルゴリズム・ルーティングに含まれるかまたは含まれないかは、図33に関して以下に説明されるように装置ビット位置及び拡張レジスタ509_j、509_kによって決定される。その動作が以下により完全に説明される、アルゴリズム・ルーティング論理回路600は、デフォルト・レジスタ509_dの内容またはルーティング表584によって供給されるターゲット・ポート識別の代わりに選択されうる3ビット・ターゲット・ポート識別をそれから生ずるべく装置ビット位置及び拡張レジスタ509_j、509_k（明確さの理由により図31には示されていない、図33参照）によって供給される6ビットDevice ID及び情報を受信する。アルゴリズム・ルーティング論理回路600は、ルータが低レベル・ルータとして構成されるならば用いられる。

【0167】マルチプレクサ594によって行われた選択は、（デフォルト・レジスタ509_fへの経路の内容の状態により）その選択、またはデフォルト・ポート・レジスタの3ビット内容を選択処理の最終段階、検査論理回路602に渡す最終マルチプレクサ599に印加される。検査論理回路602は、ターゲット・ポート選択決定のプロダクト（結果）、マルチプレクサ599の出力によって識別されるポート出力の状態を検査すべく動作する。例えば、ターゲット・ポート識別は、有効でなければならない（即ち、6または7でない）。他の検査も行われ、その一つは、識別されたポート出力が、上述したようなアクセスをシークしている特定のポート入力に対して“イネーブル”されなければならないということである。ルーティング・ループを生成することができ、かつ次いで可能なデッドロック状態を発生させることを結果として生ずる、誤りに対するバックアップとして用いられるのは、この後者の検査である。検査論理回路602は、図31に示すように、6つのポート出力502のそれぞれのポート・イネーブル・レジスタ562の内容を受信する。示したように、各ポート・イネーブル・レジスタ562の内容は、各入力ポート502に対して、どの出力ポート504が入力メッセージを送ることができ、そして勿論、どれができないかを識別する。それゆえに、例えば、ポート0がメッセージはポート3

からの送信のために送られるということを示しているDestination IDを含んでいるメッセージ・トラフィックを受信したならば、選択論理回路560は、ポート3としてターゲット・ポートを識別する3ビット量(3-bit quantity)を生じ、かつその量を検査論理回路602に印加する。更に、ポート3からのメッセージ・トラフィック送信がポート0で受信した入力メッセージ・トラフィックに対して許容されないということになったならば、ポート0に対するポート・イネーブル・レジスタ589の内容は、クロスバー論理回路500へのターゲット・ポート・アドレスの伝達(通信)をブロックする。メッセージは、クロスバー論理回路500の存在しない出力にその代わり送られ、かつ実質的に廃棄され、MPシステム18に知らせるべく誤り信号が生成される。

【0168】他方、ポート3がポート0から送られたメッセージ・トラフィックに対してイネーブルされるならば、検査論理回路602は、選択論理回路560によって発生されたターゲット・ポート識別をクロスバー論理回路500に渡し、メッセージをポート3に送らせる。検査論理回路602は、通常設計のものであり、例えば、通常ファッションで行われるべき検査及び決定を実施すべく構成された組合せ論理回路を含んでいる。上部及び下部レベルの概念的階層が可視化されたことは、少なくとも部分的にターゲット・ポート選択論理回路のコンポーネント・カウント、及びルーティング表584のサイズを制限する理由のためである。そして、ルータ14が上部または下部レベル・ルータを指定しうるし、かつサブプロセッシングシステム10A、10Bの一つまたは他の一つに配置されうることとは、その階層によるものである。ルータが上部レベルまたは下部レベル・ルータであるかは、ルーティング表584をアドレス指定するために入力メッセージのRegion IDのどの部分が用いられるかも画定する、制御論理回路509のその種々の構成レジスタに書込まれた情報によって決定されるそのルータの構成による。

【0169】これらの概念を考えに入れて、図32は、適切なポート出力へのクロスバー論理回路500を通る入力メッセージ・パケットの経路を選択するために用いられる最終ターゲット・ポート・アドレスを選択すべく用いられる決定チャート604を示す。決定チャート604は、入力メッセージ・パケットのDestination ID(及び経路選択ビットP)、及びその構成レジスタ(即ち、図31に示すレジスタ509_a, . . . , 509_h)の内容によって指定されたように、そのルータの構成に基づいて行われた決定を示す。図32が示すように、全ての決定をオーバーライドする(overriding)ことは、デフォルト・レジスタ509_fへの経路の内容である: デフォルト・レジスタ509_dの内容を選択すべくセットされたならば、全ての

他の情報(Destination ID、経路選択ビットP、他の構成レジスタの内容、等)は、不必要になる(余計になる)。上述したように、各ルータは、上部または下部レベル・ルータのいずれかとして構成される。ルータ・レベルは、Destination IDのどのビットがルーティング表584をアドレス指定すべく用いられか、かつアルゴリズム・ルーティングが用いられるべきか否かを決定する。(HiLoレジスタ509_cの内容によってそのように識別される)高レベル・ルータは、ルーティング表、クロスリンク・アドレス、またはデフォルト・アドレスのいずれかを用いる。低レベル・ルータ(HiLoレジスタ509_cがZEROを含む)は、表ベース、デフォルト、クロスリンク、及びアルゴリズム・ルーティングを用いる。

【0170】一般に、高レベル・ルータであるべく構成されたルータは、多数のルータ14を備えておりかつ多数のCPU's 12及びI/O装置を互いに伝達(通信)しているTNetリンクLを相互接続しているネットワーク“雲(clouds)”(任意ネットワーク)を相互接続すべく用いられ、大規模な並列処理(MPP)システムを形成している。他のそのようなMPPシステムが存在しうるし、かつそれは、一つのMPPシステムのそのようなネットワーク雲を他のMPPシステムに相互接続するために主に用いられる高レベル・ルータとして構成されるそれらのルータである。図27~図29をちょっと参照すると、入力メッセージ・パケットのDestination IDは、特定のポートの入力論理回路502によって受信されたときには、エラスティックFIFO 506に、かつエラスティックFIFO 506からそれが捕獲されるターゲット・ポート選択論理回路560

(図31)のレジスタ570に伝達される。メッセージ・パケットのDestination IDがそのように捕獲されるとすぐに、選択処理が始まり、適切な出力ポートに、出力ポートがイネーブルされるということ想定して、両方とも一般的に、かつメッセージ・パケットを受信している特定の入力ポートに対して、クロスバー論理回路を通してメッセージ・パケットを指向すべく用いられるターゲット・ポート・アドレスの発生に進む。

【0171】ここで図33を参照すると、3つの一に対して8ビット(8-bit to one)のマルチプレクサ620, 622, 及び624を含んでいるアルゴリズム・ルーティング論理回路600が、詳細に示されている。マルチプレクサ620, 622, 624のそれぞれの3つの選択入力(A, B, C)は、3ビット装置位置レジスタ509_jの内容、制御論理回路509に含まれる構成レジスタの別のものを受信する。各マルチプレクサ620, 622, 624の入力(0, 1, . . .)は、Device IDの6ビットの所定のものを受信する。装置位置レジスタ509_jの内容は、以下に説明する、表4の

符号化によりターゲット・ポート・アドレスとして用い
るべき3ビットの選択を制御する。MUXes 620, 622, 624によって選択されるDevice ID
の3ビットは、直接用いられない。それよりも、選択さ
れたビットは、2ビット幅レジスタ509_kの内容によ
って調整される、二つの入力ORゲート626及び3つ
のANDゲート628 (628a, 628b, 及び628c)と、制御帯状態論理回路509 (図509)に含
まれる構成レジスタの別のものとを備えている組合せ論
理回路に印加される。組合せ論理回路のプロダクト (生
成物) は、3ビット・ターゲット・ポート識別である。
幅フィールドは、ポート選択に用いる装置フィールド・
ビットの数を指定する。0 (ゼロ) の幅フィールド値

は、全てのアルゴリズム的にアドレス指定された装置
は、ポート0を通して接続することを表わす。3の幅フ
ィールド値は、アルゴリズム的にアドレス指定された装
置がいずれのポートにも接続することができることを表
わす。

【0172】装置フィールド幅拡張レジスタ509_kの
内容は、クロスバー論理回路500に適用されるターゲ
ット・ポート・アドレスを指定するために用いるビット
の選択を特定する。位置及び幅ビットの値及び意味は、
以下の表4及び表5に示される。

【0173】

【表5】

表 5

装置フィールド幅 拡張	装置の数 使用したIDビット
00	0
01	1
10	2
11	3

【0174】図9は、入力メッセージ・パケットのDe
vice IDのどのビットがMUXes 620, 622, 624のそれぞれによって選択されるかを示す。そ
れゆえに、例えば、000の装置ビット位置レジスタ5
09_jにおける(2進)値は、MUXes 620, 622, 及び624に入力メッセージ・パケットのDe
vice IDから、それぞれ、ビット2、1、及び0を選
択させる。逆に、装置ビット位置レジスタ509_jの
内容が2進100であるならば、ビット5及び4だけが、
Device IDのMUXes 620及び622によ
ってそれぞれ選択される；MUX 624の出力は、残り
のビット位置に対してZEROに強要される。装置ビ
ット位置レジスタ509_jにおける110及び111 (2
進)の値は、MUXes 620, 622, 及び624の
出力をZEROに強要させて、ターゲット・ポート0を
選択する。MUXes 620, 622, 及び624によ
ってそのように選択されたビットは、装置フィールド幅
拡張レジスタ509_kの内容により用いられる。それゆ
えに、図10が示すように、00の幅値は、MUXes
620, 622, 及び624からのビットのいずれをも
選択せず、000のターゲット・ポート・アドレスを強
要する。逆に、装置フィールド幅拡張レジスタ509_k
における10の幅値は、MUXes 620, 622, 及
び624によって選択されたビットの二つを用いる。

【0175】結果は、多くの場合、制限された値のセッ
トを有する、3ビット・ターゲット・ポート数である。
効果は、レジスタ509_kの内容によって指定された幅
を有するレジスタ509_jの内容によって指定されたビ
ット位置で始まる、3ビット・フィールドを生成するこ
とである。図27～図28及び図33にちょっと戻る

と、出力ポート504は、メッセージ・パケットを順序
付ける役割を果たす。一般に、メッセージ・パケット
は、先着順でポート出力504によって受容されかつ送
られる。しかしながら、一度メッセージ・パケットが特
定のポート出力から送信されると、多数の他のものがそ
のポート出力へのアクセスを待って遅延 (停滞) されう
る。それゆえに、アービトレーション方法は、これらの
パケット間で選択することが必要であろう。ラウンドロ
ビン・アービトレーションのような、多数の通常のアー
ビトレーション技術を用いることができる。しかしなが
ら、好ましいアービトレーション技術は、1995年6
月6日に出願され、かつ本出願の出願人にアサインされ
た、“Biased Routing Arbitration of Message traffi
c in Communication System (通信システムにおけるメ
ッセージ・トラフィックのバイアスされたルーティング
・アービトレーション)”という発明の名称であり、現
在係属中の米国出願に開示されたものである。

【0176】簡単に、各ポート出力504は、自律アー
ビター論理回路630を含む (図34)。これらのアー
ビター630は、ポート入力502のそれぞれからルー
ティング要求を取り、かつ使用されることが許可される
ポート出力の合計帯域幅の割合を表わすバイアス値を
各ポート入力502が実質的に供給されるバイアシング
技術に基づく順番でそのようなサービスを与える。この
アービトレーション技術によれば、ポート出力504の
一つに送られるべきメッセージ・トラフィックを有して
いるポート入力502は、アクセスに対するそれらの要
求を知らせる。二つ以上のポート入力アクセスをシー
クしているならば、要求されたポート出力は、それぞ
れのバイアス値を比較することによってポート入力をアー

ビットレートし、アクセスに対して一つ（例えば、最も高いバイアス値を有するポート入力）を選択する。アービトレーションに負けたポート入力は、次のアービトレーションの間にそれらの機会を増やすべくそれらの対応バイアス値が変更される；勝ったポート入力502もそのバイアス値が変更されるが、次のアービトレーションで勝つことの機会を減少するためである。ここで図34を参照すると、それからメッセージ・パケットがルータ14A（図27）によって送信されるポート出力504_nのブロック図が示されている。ポート出力504の基本コンポーネント及びそれらの機能は、：

— 入力ポートの間をアービレートすべく動作し、パケットが出力ポートによって送信される順序を決定するアービター論理回路630。

【0177】— プロトコル規則を維持しかつそれに従うために必要なときに指令記号を生成しかつ記号ストリームの中に（マルチプレクサ634を用いて）挿入すべく動作する指令記号発生器632。例えば、ルータ14Aが、受信素子が使用中なのでそれが送信することができないということを見出したときには、関連ポート出力504は、BUSY指令記号の受信に応じてメッセージ・パケット送信を停止することによって“バックプレッシャ”を課さなければならず、かつメッセージ・パケットの送信がREADY指令記号の受信によって示されるように再開することができるまでFILLまたはIDLE記号を挿入する。それが既に進行中のメッセージ・パケットを停止しなければならないならば、それは、FILL記号を送る。代替的に、BUSY指令記号が受信されたときにルータ14Aのポートが活動停止状態（メッセージ・パケットが送られない）であるならば、それは、IDLEし、かつREADY指令記号がBUSY指令記号を早めに送ったシステム素子から受信されるまでメッセージ・パケットの開始を遅延する。FILL記号は、指令記号発生器632によってポート出力504に供給される。プロトコルは、“キープアライブ”機構を実施するために出力論理回路も必要とする：ルータ14Aがまだ動作状態であることを受信素子に知らせるための記号の周期的送信（即ち、メッセージ・パケットがないときの、BUSY、IDLE）。キープアライブとして用いる記号の型は、そのときに存在している動作のモードによる。例えば、メッセージ・トラフィックがない期間中、READY記号が用いられかつ送信クロック、T_{clk}の各クロック周期またはサイクルで周期的に送られる。代替的に、ポート出力がバックプレッシャを作用させたならば、BUSY記号は、送られる。規定された時間内に、記号をまったく受信しないと、アクションに対するルータ（またはインターフェイス装置）のOLAPを介してMP18にポストされる誤りを結果として生ずる。

【0178】脳道にそれて、ルータ14によって観測さ

れるこれらのプロトコル規則は、CPU s 12（即ち、インターフェイス装置24）及びI/Oパケット・インターフェイス17によっても観測されるということを理解すべきである。そして、ルータ14AがCPU s 12A、12Bと直接的に伝達すべくシステム10（図1）にあり、かつデュプレックス・モードが用いられるときに、デュプレックス動作論理回路装置638は、CPU s 12A、12Bの一つにも接続される他のものにCPU s 12A、12Bの一つに接続されるポート出力を協調すべく用いられる。ルータ14Aのポート出力504のそれぞれは、パケット順位付けの役割を果たす。一般に、パケットは、先着順に送られる。しかしながら一度パケットが送信されたならば、複数の他のものは、待たされ続けうる。ルータ入力論理回路502のそれぞれからルーティング要求を取り、かつ上記した出願係属中の出願において示される優先度スキームに基づいて適切な順序でサービスを要求している各入力ポートに出力ポートを与えることは、各ポート出力504のアービター論理回路630の機能である。ポート出力504の各アービター630は、それが要求を与えるときに全ての他のアービター630に信号を送る。

【0179】それは、二つのCPU s 12から伝達された（デュプレックス動作における）同一記号のペアを受信する入力論理回路のクロック同期FIFO s 518である。各クロックsync 518 FIFOは、二つのCPU s 12からの記号ストリーム間で発生する遅延原因スキューに対して調整することができる。MP18にCPU 12への通信アクセスを供給したオンライン・アクセス・ポート（OLAP）がMCに含まれていたことが、上記CPU s 12の説明に関連して思い出される。MP18は、CPU 12にブート（開始）動作を終了させるべく小さなメモリ画像及びルーチンを構築するためにプロセッサ20によって実行されうるOLAP 285に命令を書込むことができた。同様なアクセスがルータ14へのMP18に供給される。図27に少しの間戻ると、ルータ14Aは、ターゲット・ポート選択論理回路の上部及び下部領域レジスタ509a、509b（図31）、及びアルゴリズム・ルーティング論理回路600（図33）の装置ビット位置及び拡張レジスタ509j、509kのような、多数の構成レジスタを含むOLAP 285'を含んで示されている。ルータ14Aを含んでいるサブシステム10Aの初期化の間中、OLAP 285'に含まれる構成レジスタは、それを一つのマナーまたは別のマナーで動作させる構成をルータ14Aに供給すべく（OLAPバス287'を介して）MP18による情報で書込まれる。

【0180】しかしながら、ルータ14Aは、OLAP 285'を通してMPへ情報（誤り表示、等）を渡す。例えば、ルータ14Aによって送られた各メッセージ・パケットは、上記したように、そのCRCが検査さ

れる。パケットのCRCが不良であるとルータ14によって決定されたならば、TPB記号でメッセージ・パケットにタグを付けることに加えて、ルータは、MP18によって後で読取ることができるOLAP285'に含まれる誤りレジスタ(図示省略)をセットすることによってMP18にフラグをたてる。それゆえに、システムは、この特徴(機能)を通して送信フォルトを知らせる手段を備えている。

【0181】クロッキング：明らかに、CPU s 12がデブレックス・モードで同期的にマッチしたペアとして適切に動作されるならば、それらが用いるクロック信号は、同期されていなければならない。図36は、クロック発生回路設計を示す。同期を維持するために各サブプロセッサ・システム10A/10Bに一つのクロック発生器回路が存在する。参照番号650で一般に指定される、クロック発生器回路は、クリスタル発振器回路652a及び通減係数8(divide-by-eight)の計数回路(カウンタ)652bを備える発振器回路652を含む。クリスタル発振器回路652aは、25/16 5 Mhzの周波数を有するマスタ・クロック(M_CLK)信号を生起すべく8によって分割される12.5 Mhzの周波数を有する周期的信号を生成する。M_CLK信号は、SYNC_CLKにも印加される。クロック発生器654に印加されて、M_CLK信号は、M_CLKに対して全てが位相封じ込みされた、多数の50 Mhzクロック信号を生起すべく用いられる。これらの50 Mhz信号は、クロック回路650を含んでいるサブプロセッサ・システム(例えば、10A)の種々の素子(例えば、CPU12、ルータ14、等)に分配されかつ用いられる。

【0182】クロック発生器654は、M_CLK信号を受信しかつ、それ自身の位相-封じ込みされた複製である、フィードバック・クロック信号と比較すべく接続された位相コンパレータ660を含んで示される。M_CLKとフィードバック・クロック信号との間の位相差を表わすアナログ電圧(V)である、位相コンパレータ回路660の出力は、位相及び周波数の両方において、M_CLK信号に対してクロック発生器によって生成された50 Mhz信号のロックを維持すべく電圧制御型クリスタル発振器(VCXO)662に印加される。位相コンパレータ660が、M_CLKとフィードバック信号の間で所定の位相範囲よりも大きい位相差を検出したならば、それは、位相封じ込み(phase lock)の損失を示すべくLOCK信号をディアサートする。VCXO662(図36)は、厳しい許容誤差範囲(tight tolerance)内で動作すべく構成された100 Mhz電圧制御型クリスタル発振器である。VCXO662の生成物は、50 Mhz信号を生成すべく2で、かつM_CLK信号の複製、フィードバック信号を生成すべく64で、VCXO662の出力をカウント・ダウン(分割)する同期カ

ウンタに印加される。カウンタ663によって生成された50 Mhzクロック信号は、サブプロセッサ・システム中の必要な場所に分配される。

【0183】ここで、図37を参照すると、周波数封じ込み動作の一对のサブプロセッサ・システム10A、10B(図1~図3)に対する同期クロック信号を生起するために用いられる二つのクロック回路650の相互接続及び使用を示す。図37に示すように、サブプロセッサ・システム10A、10Bの二つのCPU s 12A及び12Bは、それぞれ、発振器回路652A、652Bを含んでいる、クロック回路650A及び650Bとして図37に示す、クロック回路650を有する。しかしながら、CPU s 12の一つのクロック発振器652だけが、両方のCPU s 12に対するM_CLK信号を生起するために用いられる。図37は、両方のCPU s 12のクロック発生器654A及び654Bを駆動するために用いられるCPU12Aの発振回路652Aを示す。ドライバ及び信号回線667は、サブプロセッサ・システム10Bのクロック発生器654Bに発振器回路652Aによって生起されたM_CLK信号を送付すべく二つのサブプロセッサ・システムを相互接続する。フォルト分離に対し、かつ信号の品質を維持するために、M_CLK信号は、個別ドライバ及びループバック接続668を通してサブプロセッサ・システム10Aのクロック発生器654Aに送付される。ループバック接続668の理由は、信号相互接続667によって課せられた遅延によりクロック発生器654Bによって見られるものにおおよそし等しい遅延を発振器回路652Aとクロック発生器654Aとの間に課すことである。

【0184】簡略化のために、図37に特に示されないのは、発振器回路652が発振器652Aからのそれらをミラーする接続及びドライバを有するということである。それは、どの発振器回路652A、652Bが二つのクロック発生器654A、654Bを駆動する発振器であるかを制定するCPU s 12A、12Bを接続するための用いるケーブルである。即ち、一つの方法(one way)で接続されて、ケーブル(図示省略)は、サブプロセッサ・システム10A、10B間で図37に図示した接続を制定する；別の方法で接続されて、接続は、同じであるが、発振器652Bが用いられる発振器である。図37を続けると、サブプロセッサ・システム10Aの発振器回路652Aによって生成されたM_CLK信号は、それらの対応SYNC_CLK信号及びクロック発生器654A、654Bによって生成された50 Mhz信号から生起された種々の他のクロック信号のように、両方のサブプロセッサ・システム10A、10Bによって用いられる。それにより、ペアになったサブプロセッサ・システム10A、10Bのクロック信号は、デブレックス・モードに必要な周波数封じ込み動

作に対して同期される。

【0185】クロック発生器654A, 654BのVCOs662は、通常設計のものであり、かつ位相コンパレータ660からの印加されたアナログ電圧(V)が、制御限度の外側である(位相コンパレータ660から受信したクロック信号がひどく位相がずれていることを示している)ときでさえも所望の周波数を維持し続ける型のものである。これは、サブプロセッサ・システムがもはや周波数封じ込みされていないけれども、両方のクロック発生器654A, 654Bに、発振器回路652Aの不適切な動作のフェース(face)において二つのサブプロセッサ・システム10A, 10Bにクロック信号を供給することを継続させる。クロック発生器回路654A, 654Bの(M_CLKが存在しかつその複製、フィードバック信号と同期(sync)であることを示している)位相コンパレータ660によってアサートされたLOCK信号は、両方ともに誤り論理回路670A, 670Bに結合される。LOCK信号をアサートすることは、クロック発生器654によって生成された50MHz信号が、M_CLK信号に、位相及び周波数の両方において、同期されることを意味する。それゆえに、LOCK信号のいずれかがZEROであった(即ち、ディアサートされた)ならば、誤り論理回路670は、どのクロック発生器がそのLOCK信号をディアサートしたかを決定しかつOLAP285を介してMP18に知らせる。両方のLOCK信号がディアサートされたならば、CPUsは、クロック発生器654A, 654Bを駆動している発振器回路652Aが正しく動作していないことをそれから想定することができる。

【0186】— 一定比率クロッキング: 上述したように、一対のデュプレックスされたCPUs12とルータ14A, 14B(図1〜図3)の間の記号転送は、周波数封じ込みモードでそのように行われる; 即ち、記号ストリームを付随し、かつ受信素子(ルータ14、またはCPU12)クロック同期FIFOに記号をプッシュする(押す)ために用いられるクロック信号は、クロック同期FIFOsから記号をプルする(引く)ために用いられる受信素子のものに、位相でなければ、周波数において実質的に同一である。例えば、ルータ14Aから一対のデュプレックスされたCPUs12A, 12Bに送られた記号を示す、図35を参照すると、ルータ14Aで発生している(かつ受信クロック(Rcv Clk)としてCPUs12A, 12Bで受信されるべき、記号ストリームを付随している、)クロック信号は、局所クロック(Local Clk)に周波数において実質的に同一である。前者(Rcv Clk)は、各CPUのクロック同期FIFOs126に記号をプッシュするために用いられ、後者は、FIFOsから記号をプルするために用いられる。この技術は、同じ周波数のものであり、クロック信号(T_Cl k/Rcv Cl k及びLo

cal Clk)に対してよく動作し、かつ偶然にもTNetリンクL上の通信に用いられたクロック周波数である。しかしながら、送信媒体、即ち、TNetリンクLの電氣的または他の特性に従うために、その媒体にわたり記号を送信するために用いられるクロック信号の周波数が制限されが、受信エンティティが制限されないと仮定したならば、ここでCPUs12は、より高い周波数のクロック信号で動作することができる。そのようが状況において、それぞれのクロック同期FIFOs126からプルされた記号に対して同期が二つのCPUsの間で維持されるということを確実にするために準備がなされなければならない。

【0187】ここで、一定比率クロッキング機構が、二つのクロック同期FIFOs126の動作を制御するために用いられ、それらがFIFOsにプッシュされると同じ速度で二つのFIFOsから記号をプルするクロック信号を供給する。図38を参照すると、参照番号70で示された一定比率クロック制御機構が示されている。図38が示すように、クロック同期FIFO制御機構700は、その並列出力がN対1マルチプレクサ(MUX)704に印加されるプル・セット可能なマルチ・ステージ直列シフト・レジスタ702を含む。直列シフト・レジスタ702は、シフト・レジスタのクロック(CK)入力に印加されるより速い(より高い周波数)局所クロック信号(Local Clk)で動作される。15ビット・バス701は、直列シフト・レジスタ702をプリセットすべくデータ入力(DI)にプリセット(PR₁)を運ぶ。直列シフト・レジスタを形成しているステージ数は、わかるように、局所的に用いられるクロック信号の周波数に対する記号が伝達されかつクロック同期FIFOs126にプッシュされるクロック信号の比率により、あらゆるものでありうる、ということが当業者には明らかであろう。ここでは、15ステージが十分であると思われる。

【0188】MUX704は、シフト・レジスタ702から15並列データ出力(DO)の一つを選択すべく動作し、クロック同期FIFOs126から記号をプルし、かつプル・ポインタ・カウンタ130を動作(更新)すべくLocal Clk信号として用いられる一定比率クロック制御機構の、出力として、MUXの入力(I)に印加される。選択された出力は、MUXの出力(O)からも結合されかつ直列シフト・レジスタのシフト・イン(SI)入力に印加される。選択は、4ビット・カウンタで実施されうる。— サイクル長論理回路のデータ入力(DI)に印加された(4ビット)プリセット(PR₂)値でプリセット可能なサイクル長論理回路706によってなされる。サイクル長論理回路の4ビット出力は、MUX704の選択(C)に印加される選択値を形成する。本質において、一定比率クロック制御は、所定の時期(期間)にわたりRcv Clkで同じ数

のクロック・エクスカージョン(excursions)を有している出力信号を生成すべく動作する。クロック同期FIFO 126に記号をプッシュするために用いられるクロック信号に対するCPU 12のクロック信号の間のN:M (ここで $N > M$)の比率が、Rcv Clkであると想定すると、直列シフト・レジスタは、シフト・レジスタのMステージが第1のデジタル状態(例えば、ONE)を保持し、かつ他が別のデジタル状態(例えば、ZERO)を保持するようにプリセットされる。サイクル長論理回路は、(もちろん、最後または15番目のステージが第1のステージへのフィードバックを形成するような、Mが15であることを除き)Mステージを有するトランケートされた直列シフト・レジスタを、実質的に、生成する直列シフト・レジスタの出力を選択すべく値でプリセットされる。例は、これをよりクリアにする。

【0189】図35をちょっと参照して、記号が50Mhzクロックでルータ14Aから二つのデュプレックスされたCPU s 12に送信されると想定する。それゆえに、記号は、50Mhz速度でCPU sのクロック同期FIFO s 126にプッシュされる。CPU sのクロック信号が40Mhzであると更に想定する。従って、Rcv Clk信号に対する局所クロック(80Mhz)の比率は、8:5である。直列シフト・レジスタは、15ステージの最初または第1の8つが5つのONE sと3つのZERO sを含むようなビット・パターンでプリセットされる。サイクル長論理回路は、MUX 704による直列シフト・レジスタの8番目のステージの選択を動作する値でプリセットされる。それゆえに、シフト・レジスタ及びサイクル長論理回路は、それぞれが100ns(ナノ秒)期間である3つの“待ち(wait)”状態及び5つの“出る(out)”状態を、実質的に、含んでいる8つのステージを有する直列シフト・レジスタを、実質的に、生成する値が供給される。従って、クロック同期FIFO s 126から記号をプルするクロック信号を生成する、MUX 704の出力、Rcv Clkは、各100ns期間に対して、5つのクロック・パルスを含む。それゆえに、各100ns期間に対して、5つの記号がクロック同期FIFO s 126にプッシュされ、かつ5つの記号がクロック同期FIFO s 126からプルされる。

【0190】この例は、図39に記号的に示され、図40に示されたタイミング図は、制御論理回路700の動作を示す。各100ns期間に対して、Rcv Clkの5つのクロック・パルス(図40において“IN”と示された)は、クロック同期FIFO s 126に記号をプッシュする。それと同じ100ns期間中に、直列シフト・レジスタ702は、MUX 704によって選択されたステージ710を通して“01101011”シーケンスを循環させ、Rcv Clk信号と同じ数のアク

ティブ・クロック・パルスを有するLocal Clk信号を生成する。シフト・レジスタ702のステージの数は、ここの示されたようなシステムにおける最も一般的なクロック・スピード差分を収容すべく変更されうるということが当業者には明らかである。好ましくは、シフト・レジスタ702は、示されたように、15のステージを有し、比較的広い範囲のクロック比率をカバーするための能力(機能)を供給する。ここで理解できるように、この一定比率クロッキングのこの技術は、決して1クロック以上ずれない。更に、例えば、5つのクロックをカウントしかつ追加の記憶(例えば、同期FIFOのサイズにおける増加)を必要としかつより待ち時間を課す3つのクロックを保持するよりも良好な実施である。

【0191】ここに示された一定比率クロック回路(図38及び図39)は、一周波数のクロック様式(regime)から異なる、より高い周波数のクロック様式にデータ素子を転送するために用いられる。クロック同期FIFOの使用は、二つの異なるソースから同一の指令/データ記号のペアを受信すべく同期化された、デュプレックスド・モードで動作するとき、信号遅延の影響を補償するためにここで必要である。しかしながら、ここに開示された一定比率クロック回路は、クロック同期FIFOの代わりに少なくとも二つのレジスタが存在する限り、二つの異なるクロック様式間でデータを伝達するために有用であるというのが当業者に自明である。より高い周波数クロック様式からより低い周波数クロック様式にデータを転送することは、一定比率クロック回路702によって生起されたクロック信号の制御下で入力ステージまたはレジスタにデータ素子を転送すべく一定比率クロック回路702を用いる;より低いクロック様式のクロック信号は、二つ(または、ここでは、それ以上)の受信レジスタ・ステージの間でデータ素子を転送し、かつそれからデータ素子を除去するために用いられる。逆に、より低い周波数クロック様式からより高い周波数を有しているものに転送されたデータ素子は、ここに示されたように本質的に動作する。

【0192】この概念は、異なるクロック信号が用いられるところに用いることができる。例えば、マイクロプロセッサ技術においてよく知られるように、多くのマイクロプロセッサは、一周波数のクロック信号に応じて動作するマイクロプロセッサが、異なる、通常より低い周波数のクロック信号に応じて動作する同期装置(例えば、メモリ、または外部、システム・バス)と伝達するときに“待ち”状態を挿入すべく構成されている。一般に、そのようなマイクロプロセッサ/装置通信は、より遅いクロック信号がマイクロプロセッサ・クロック周波数の整数倍であるということが要求される。一定比率クロック制御回路702は、広い範囲の可能なクロック比率を供給できる。

【0193】I/Oパケット・インターフェイス：サブプロセッサ・システムが局所I/Oを有することを必要としないように他のサブプロセッサ・システムのI/Oが利用可能であるということが考えられるけれども、サブプロセッサ・システム10A、10B、等のそれぞれは、種々の周辺装置で実施される、入力/出力機能を有する。あらゆる場合において、局所I/Oが供給されたならば、周辺装置及び/又はMP18は、I/Oパケット・インターフェイス16を介して伝達（通信）する。I/Oパケット・インターフェイス16は、装着されたI/O装置に対してより互換性があるかまたはネーティブな形にそれがTNetリンクLから受信した入力メッセージ・パケットを変換すべく動作する；次いで、I/Oパケット・インターフェイス16は、反対方向にも変換して、装着されたI/O装置から“ネーティブI/O”（NIO）を受信し、上述した8B-9Bフォーマットのデータにバイトを符号化し（上記、表1参照）、かつデータを宛先に送るために必要なパケットを形成する。更に、特定のI/O装置（例えば、信号回線）に対して最も普通の方法でアサートされる、I/O装置からの割込みは、I/Oパケット・インターフェイス装置16によって受信され、かつそれが上述したように処理される、割込みが意図したCPU12に送られる割込みパケットを形成するために用いられる。それゆえに、NIOバス上の装置は、読取りをし、書込みをし、かつCPU12のメモリ28にTNetリンクL及びルータ14を通して透過的に渡されたデータ/制御情報で慣例的にメッセージ・パケットを介して割込みを発行する。

【0194】そして、I/Oパケット・インターフェイス16が、I/O装置の一つ、MP18として、それに接続されうるけれども、I/Oパケット・インターフェイス16は、構成情報を、OLAPバスを介して、受信するMC26（図17B）に含まれ（OLAP285）かつルータ14に含まれた（OLAP285'；図27）ようなOLAPも含む。

【0195】オン・ライン・アクセス・ポート：MP18は、（IEEE1149.1-1990、5月、1990年、SH13144、Institute of Electrical and Electronic Engineers, 345 East47th Street, New York, NY 10017に基づく）IEEE標準1149.1に準拠するインターフェイス信号でインターフェイス装置24、メモリ・コントローラ（MC）26、ルータ14、及びI/Oパケット・インターフェイスに接続する。OLAP258は、そのIEEE標準を実施し、かつOLAP258の構成及び動作は、それがどの素子（例えば、ルータ14、インターフェイス装置24、等）を用いたかに関係なく本質的に同じである。図41は、IEEE1149.1標準インターフェイスを実施しているOLAP258の一般構造を図式的に示す。O

LAPは、ここに説明した素子のあるものを実施するために用いられる各集積回路チップ上で形成されるのが好ましい。例えば、各インターフェイス装置24、メモリ・コントローラ26、及びルータ14は、OLAPも含むアプリケーション指定集積回路（ASIC）によって実施され、ASICの回路素子へのアクセスをMP18に供給する。それゆえに、図41に示したOLAP158の説明は、インターフェイス装置24、MC26、及びシステムのルータ14に関連するOLAPを説明する。

【0196】図41に示すように、直列バス19Aは、4つの1ビット信号回線を備えている：OLAP258へ周期的クロック信号を運ぶ試験クロック（TCK）信号回線；2状態指令信号を伝達する試験指令（TCM）信号回線、OLAPヘデータを運ぶ試験データ・イン（TDI）信号回線；及びOLAPからデータを伝達する試験データ・アウト（TDO）信号回線。これらの信号は、IEEE1149.1標準の要求事項に従っている。OLAP258は、直列バス19AのTCK及びTCM回線上で受信したクロック及び指令信号に応じてOLAPの動作を制御する4ビット状態マシンを含む。OLAP258によって受信されたデータ（及び/又は命令）は、16ビット命令レジスタ（IR）802及び/又は32ビット・データ・レジスタ（DR）104によって記憶される；データは、DR804だけが関連論理回路（例えば、ルータ14）からのデータで装填することができるということを除き、IR、DRレジスタのいずれかから伝達されうる。OLAP258に関連するが、その一部でないものは、MP18（OLAP258を介して）及びOLAP258が関連される論理回路の両方によってアクセスすることができ64の32ビット・レジスタまでを含むレジスタ・ファイルの形の構成レジスタ806である。例えば、構成レジスタ806のレジスタのあるものは、ルータ14の制御及び状態論理回路509（図27）を形成する。構成レジスタ806は、IR802によって最初に供給される10ビット命令によって命令される（32ビット）位置（即ち、64の利用可能な32ビット・アドレスの選択されたもの）でDR804から書込まれる。構成レジスタ806を装填するための命令は、命令復号論理回路810によって復号される4ビット部分を含み、指令発生器812に印加された合成復号は、読取りまたは書込み動作を識別する。動作の目的、即ち、読取られるかまたは書込まれるべき、構成レジスタ806を構成している64レジスタの一つは、アドレス復号論理回路814によって復号される6ビット・アドレスによって識別される。指令発生器812は、状態マシン800の状態も受信する。それゆえに、状態マシン800によって想定された特定状態により、命令復号論理回路810からの復号された指令と一緒に、書込みまたは読取り指令信号は、構成レジ

タ806に、(アドレス復号論理回路814によって復号されたような)命令の6ビット・アドレスによって識別された64のレジスタの一つで読取りまたは書き込みを実行させるべく指令発生器論理回路812によって生成される。

【0197】MP18(図1)によって供給されたデータは、マルチプレクサ816を介してDR804に書込まれる。OLAP258を用いている論理回路は、二つの個別ソースからDR804を書込みうるし、IR802に早めに書込まれた命令情報及び直列バス19AのTCK及びTCM信号回線によって運ばれた信号(signalling)による状態マシン800の動作を用いてMP18によってDR804に選択的に結合されかつ書込まれたそれらのソースで32ビット・レジスタを供給する。32ビットDR804は、適切な1149.1命令の使用と一緒に、“CAPTURE-DR”、“SHIFT-DR”、及び“UPDATE-DR”として1149.1に記載された状態を通して状態マシン800をステップすることによってアクセスされる。命令の追加ビットは、DR804に、CAPTURE-DR状態によりチップ状態情報を含んでいるチップ内の選択された値を読取らせる。他の1149.1命令は、構成及び初期化目的のためにUPDATE-DR状態に選択されたレジスタに対してレジスタ内容をコピーさせる。DR804の内容は、1149.1 SHIFT-DR状態を用いて(直列バス19Aを介して)MP18と変換される。OLAP258の構成及び動作の更なる情報に対して、IEEE1149.1標準(IEEE1149.1-1990、5月、1990年、SH13144)が参照される。

【0198】**非対象変数**：“非対象変数”は、他のものから一対のCPU12の一つにおいて異なるか、または異なりうる、値である。非対象変数の例は、他のCPUのものから異なる、CPU-読取り可能位置、例えば、レジスタ外側メモリ28か、または訂正可能メモリまたはキャッシュ誤りの発生をトラックするために用いられるレジスタの内容、に割り当てられかつ保持される通し番号を含むことができる(誤りを検出し、訂正し及び報告することは、デュプレックスされたCPUにロックステップ同期を失わせないものと想定する)。デュプレックス・モードでは、非対象変数の注意深い処理は、論理的の同等であると仮定して、(各CPU12のメモリ28に維持された)システム・メモリの多重コピーが、常に同一のデータを含むことを確実にするために重要である。非対象変数が二つのデュプレックスされたCPU12のそれぞれによって単に読取られたならば、次いでメモリへ書込まれ、各CPUのメモリ28の内容は、それにより少なくともそれぞれによって読取られた値で異なる。一対のCPU12を許容するために、デュプレックス・モードで動作し、非対象変数を処

理すべく、“ソフト・ヴォート(soft-vote)”(SV)論理回路素子900(図43)が各CPU12の各インターフェイス装置24に供給される。図43～図44が示すように、各インターフェイス装置24のSV論理回路素子900は、バス回線902a及び902bを含んでいる、2ビットSVバス902によって互いに接続される。バス回線902aは、CPU12Aのインターフェイス装置24からCPU12Bのそれらに1ビット値を運ぶ。逆に、バス回線902bは、CPU12BのSV論理回路素子900からCPU12Aのそれらに1ビット値を運ぶ。

【0199】図44に示すのは、CPU12Aのインターフェイス装置24aのSV論理回路素子900aである。各SV論理回路素子900は、示さない限り、論理回路素子900aの説明が(CPU12A、インターフェイス装置24bの)他の論理回路素子900a、及び(CPU12Bの、インターフェイス装置24a、24bの)900bにも同様に適用されるということが理解されるべきであるように各他のSV論理回路素子900に対して構成及び機能において実質的に同一である。図44が示すように、SV論理回路素子900aは、4つの1ビット・レジスタを含む：出力レジスタ904、局所入力レジスタ906、遠隔入力レジスタ907、及び出力イネーブル・レジスタ912。出力レジスタ904は、共有バス回線902aに、マルチプレクサ(MUX)914及び3状態ドライバ918を介して、結合される。CPU12Aの論理回路素子900aだけが、バス回線902aを駆動し、次いで二つの論理回路素子に一つだけがバス回線を駆動する。どれか一つは、イネーブル・レジスタ912の内容に依存する。従って、バス回線902aは、論理回路素子900aの出力レジスタ904を、CPU12Bの論理回路素子900bのそれぞれの遠隔入力レジスタ907に伝達する。バス回線902aは、論理回路素子900aの一つの(マルチプレクサ914及びドライバ918を介して)出力レジスタ904を、論理回路素子900aの他の(並びにそれ自身の)局所入力レジスタに伝達する。この方法で、CPU12Aの二つのインターフェイス装置24a、24bは、互いに非対象変数を伝達することができる。

【0200】同様なファクションで、CPU12Bの論理回路素子900bの出力レジスタ904は、論理回路素子902a(及び他のインターフェイス装置24bのもの)の遠隔レジスタ907にバス回線902bによって伝達される。論理回路素子902は、一対の構成レジスタ74(図9)を形成する。それゆえに、それらは、出力レジスタ904及び/又はイネーブル・レジスタ912を選択しかつ書込むためか、または入力局所及び遠隔レジスタ906及び907を選択しかつ読取るために少なくともアドレス/データ・バス74(図44にバス74'として示す)の一部にわたる必要なデータ/アド

レス情報を伝達することによってプロセッサ装置20によって書込まれる。MUX914は、SV論理回路素子900aに対するバス回線902aの選択的使用をCPU12Aの各インターフェイス装置24に供給するか、または一対のCPU12をロックステップ、デブプレックス動作に至らせるために用いれる（以下に説明する）再統一処理の間中に遭遇したならばBUS ERROR信号を伝達するために動作する。出力イネーブル・レジスタは、3状態ドライバをイネーブルする（またはディスエーブルする）ビットで書込まれ、それがSV出力レジスタ904の内容でバス回線902aを駆動する。

【0201】上述したように、SV論理回路素子900は、デブプレックス・モードで動作しているときにCPU12A、12Bに、非対象変数のビット毎の交換を実施される。CPU12A、12Bがデブプレックス・モードであるときには、それらは、両方とも、もし同じ瞬間でなければ、実質的に同じ仮想瞬間(virtual moment in time)に同一命令ストリームの同じ命令を実行していることを思い出す。それらの間の非対象変数の交換は、次の通りである。両方のCPUは、命令ストリームに応じて、かつ本質的に同時に、各CPUの両方のインターフェイス装置24の論理回路素子900のイネーブル・レジスタ912を書込む。各CPUの二つの論理回路素子900の一つは、関連ドライバ916をイネーブルする状態で書込まれる；他は、高いインピーダンス状態にドライバの出力を置く状態で書込まれる。それは、関連ドライバ916をイネーブルすべく書込まれる両方のCPU12A、12Bのインターフェイス装置24aの論理回路素子900と関連した出力イネーブル・レジスタ912であるということを想定する。それゆえに、各CPUのインターフェイス装置24aの出力レジスタ904は、バス回線902に伝達される；即ち、インターフェイス24a（CPU12A）の論理回路素子900aに関連した出力レジスタ904は、バス回線902aに伝達され、CPU12Bのインターフェイス装置24aの論理回路素子900bに関連した出力レジスタは、バス回線902bに伝達される。CPU12は、両方とも、それらの対応出力レジスタ904に非対象変数のビットを書込み、それぞれの関連遠隔入力レジスタ907の最大クロック・スキューを許容した後、読取りが後に続く。出力レジスタ904は、各CPUによって再び書込まれ、遠隔入力レジスタ907を読取ることが再び後に続く。この処理は、変数全体が各CPU12の出力レジスタ904から他の遠隔入力レジスタに伝達されるまで、一度に1ビット、反復される。CPU12Bの両方のインターフェイス装置24は、非対象情報のビットを受信することに注目する。

【0202】ソフトウォーク機構の使用の一例は、通し番号(serial numbers)の交換である。構成レジスタ7

4の一つは、互いにデブプレックスされう二つのCPUのそれぞれを識別し、かつ互いにそれらを区別すべく始動(スタート・アップ)でセットされる1ビット・レジスタ(図示省略)である。それゆえに、一つのCPUの1ビット・レジスタは、他のCPUのものから異なる状態にセットされる。これは、そのCPUに割り当てられた通し番号で装填される他の構成レジスタで、スタート・アップの間中も、続けられうる。通し番号に対して構成レジスタのどれが装填されたかは、1ビット識別レジスタの状態による。それゆえに、二つのCPUは、それを一つのCPUにおいてそれ自身の通し番号を有する“R1”(図示省略)と呼ぶその一つのレジスタを除き、それぞれそれらの通し番号を含んでいる二つの同一レジスタを有し、他のCPUは、構成レジスタ“R2”(図示省略)にそれ自身の通し番号を有する。これらの値がデブプレックスされたCPUによってメモリに書込まれることができる前に、R1、R2構成レジスタは、ソフトウォーク機構を用いて、“調和”されなければならない。SV論理回路素子900は、説明される再統一処理の間に発生しうるバス誤りを伝達するためにも用いられる。再統一が実行されるときには、REINT信号がアサートされる。図44に示すように、REINTは、MUX914の制御(C)入力に印加される。それゆえに、REINTがアサートされるときには、BUS ERROR信号がMUX914によって選択されかつバス回線902aに伝達される。

【0203】同期：個別に動作している(シンプレックス・モード)か、またはペアになりかつ同期ロックステップ(デブプレックス・モード)で動作しているサブプロセッサ・システム10A、10Bの適切な動作(図1及び図4)は、CPU12A、12Bとルータ14A、14Bの間で伝達されるデータが適切に受信され、かつ(CPU12A、12Bの；図9)クロック同期FIFOs102及び(ルータ14A、14Bの；図29)クロック同期FIFOs519がデータまたは指令として誤って解釈されないといふことの確保が必要である。種々のクロック同期FIFOs(CPU12の)102及び(ルータ14の)518のプッシュ及びプル・ポインタは、少なくとも近周波数動作に対して初期化されることが必要である。一般に、これは、パワーが最初に印加され、プッシュ及びプル・ポインタ・カウンタをある規準距離離してセットし、ある既知の状態に関連FIFOキューをプリセットするときに、パワー・オン信号(図示省略)によって通常のファクションで行われる。これがなされて、全てのクロック同期FIFOsが近周波数動作に対して初期化される。それゆえに、システム10が最初にオンラインにされた(パワー・アップされた)ときに、CPU12A、12Bとルータ14A、14Bとの間の通信リンクの動作は、近周波数モードである。

【0204】しかしながら、CPU s 1 2 A, 1 2 B がデュプレックス・モード動作にスイッチされるときに、更なるものが要求される。まず、各TNetリンク上のCPU s 1 2 A, 1 2 B とルータ 1 4 A, 1 4 B との間にデータを送付するために用いられるクロッキングは、周波数封じ込み動作にスイッチされなければならない。次に、デュプレックス・モード動作のロックステップ動作を適切に実施するために、クロック同期FIFO s は、別の経路に見出されない一つの経路における遅延を収容するためにそれからそれらがデータを受信する特定ソースで動作すべく同期されなければならない。例えば、デュプレックス・モード動作は、ペアになったCPU s 1 2 が同じ仮想時間に同一命令ストリームの各命令を実行することを必要とするということを思い出す。

(“仮想”時間とは、ペアになったCPU s 1 2 による同一命令の実際の実時間実行が、少しの量だけ異なっても、外側世界によって見られるときにそれらのアクションは、まったく同じであるということを意味する。) ルータ 1 4 A 及び 1 4 B からの入力データは、ロックステップ動作のコンテキストにおいて、ほとんど同時に二つのCPU s によって受信されなければならない。ルータ 1 4 A, 1 4 B の一つまたは別のものからCPU s 1 2 A, 1 2 B への通信経路における遅延は、考慮されなければならない。それは、メッセージ・パケット記号を受信し、通信経路において課せられうる遅延に対して調整し、デュプレックス・モード動作に必要なロックステップ同期を維持すべく同期マナーで二つのCPU s に記号を与えるべく動作するペアになったCPU s 1 2 のクロック同期FIFO s 1 0 2 である。

【0205】同様なファッションで、CPU s 1 2 の一つからルータ 1 4 A, 1 4 B によって受信された各記号は、(以下に、更に説明する) CPU s の可能な発散を検査するために他からのものと比較されなければならない。それは、二つのCPU s 1 2 から受信した記号がクロック同期FIFO s から同時に検索されるように通信経路における遅延を収容すべく調整するCPU s 1 2 からメッセージ・パケットを受信するルータ 1 4 A, 1 4 B のクロック同期FIFO s 5 1 8 の機能である。CPU s 及びルータのクロック同期FIFO s がどのようにリセットされ、初期化され、かつ同期されるかを説明するまえに、同期ロックステップ・デュプレックス・モ

ード動作を維持するためのそれらの動作の理解は、有用であると信ずる。それゆえに、図 3 5 をちょっと参照すると、例えば、ルータ 1 4 A からデータを受信するCPU s 1 2 A, 1 2 B のクロック同期FIFO s 1 0 2 が示されている。図 3 5 は、次いで、ルータ 1 4 A からのデータ/指令記号及びクロックを二つのデュプレックスされたCPU s 1 2 A, 1 2 B にそれぞれ結合する、1 0 ビット・バス 3 2_x 及び 3 2_y に接続されたルータ 1 4 A のポート出力 5 0 4₄ 及び 5 0 4₅ を示す。メッセージ・パケットがCPU 1 2 を識別している単一宛先アドレスを有しうるけれども、パケットは、記号毎に、ルータ 1 4 A によって複製され、かつ両方のCPU s 1 2 A 及び 1 2 B の実質的に同時に送信されるということを思い出す。

【0206】二つのCPU s 1 2 A, 1 2 B は、ルータ 1 4 A からCPU s の一つ (例えば、CPU 1 2 B) によって受信された記号が、他のCPU (CPU 1 2 A) による (ルータによって複製されたような) 同一記号のそれらの受信に関して未知 (しかし最大) 量の遅延を経験するように配置されるであろう。この遅延は、ルータ 1 4 A からCPU 1 2 B へ、記号及び付随送信機クロックT_C1kを伝達するバス 3 2_y の 6 4 0 で表される。デュプレックス動作の間中に同一記号ストリームを受信するためのクロック同期FIFO s 1 0 2_x, 1 0 2_y の動作を考える。以下の、表 6 は、その動作を示す。簡単のために、表 6 は、遅延 6 4 0 が送信クロック (T_C1k) の二期間以上でないということを想定する。しかしながら、遅延 6 4 0 がT_C1kの二クロック時間以上であるならば、キュー 1 2 6 の深さは、プッシュ及びプル・ポインタ・カウンタ 1 2 8 及び 1 3 0 の内容の間で増加した距離を供給すべくそれゆえに増加されなければならない。例えば、遅延 6 4 0 が記号のCPU 1 2 B での到着が、CPU 1 2 A での同じ記号の到着よりも 3 T_C1k 期間だけ大きいようであるならば、プッシュ及びプル・ポインタ・カウンタ間の距離は、少なくとも 4 であるべきである。それゆえに、そのような場合における、キュー 1 2 6 の深さは、6 記号位置、またはそれ以上である。

【0207】

【表 6】

表 6

項 目	RST	clk	clk	clk	clk	clk	clk	clk
		1	2	3	4	5	6	7
CPU 12A 値								
プッシュポインタ	0	1	2	3	0	1	2	3
プルポインタ	2	3	0	1	2	3	0	1
バイト 0	IDLE	A	A	A	A	E	E	E
バイト 1	IDLE	IDLE	B	B	B	B	F	F

バイト 2	IDLE	IDLE	IDLE	C	C	C	C	G
バイト 3	IDLE	IDLE	IDLE	IDLE	D	D	D	D
アウト reg	IDLE	IDLE	IDLE	A	B	C	D	E
CPU 12B 値								
プッシュポインタ	0	0	1	2	3	0	1	2
プルポインタ	2	3	0	1	2	3	0	1
バイト 0	IDLE	IDLE	A	A	A	A	E	E
バイト 1	IDLE	IDLE	IDLE	B	B	B	B	F
バイト 2	IDLE	IDLE	IDLE	IDLE	C	C	C	C
バイト 3	IDLE	IDLE	IDLE	IDLE	IDLE	D	D	D
アウト reg	IDLE	IDLE	IDLE	A	B	C	D	E

【0208】表6の上半分は、CPU 12A（インターフェイス装置24a）に対するプッシュ及びプル・ポインタ・カウンタ128，130によって保持される値、キュー126の4つの記憶位置（バイト

0，．．．，バイト3）の各々の内容、初期リセット

（RST）期間及び送信機クロック、T_{CLK}の後続クロック・サイクルに対する出力レジスタ132の内容を示す。表6の下半分の行は、複製された記号ストリームの各記号に対するCPU 12Bインターフェイス装置24aのFIFO102_yに対して同じものを示す。遅延640が2T_{CLK}期間以上でない想定して、

（カウンタ128，130で維持される）プッシュ及びプル・ポインタは、二位置（two locations）離れたキュー126の位置を指し示す。プッシュ・ポインタ・カウンタ128は、それぞれ受信した記号が記憶されるキュー126の次の位置を指し示し、かつプル・ポインタ・カウンタ130は、それぞれ記号がキューからプルされる位置を指し示す。表6、図35を参照して、“IDLE”記号のストリームをそれが先に送っているプロトコルに付いている、ルータ14Aは、記号Aで始まる、記号ストリーム（メッセージ・パケット）を送り始める、ということをごここで想定する。表6が示すように、記号Aは、遅延640によりCPU 12Aでのその到着よりも一サイクル後でCPU 12Bに到着する。しかし、CPU 12Bに対するプッシュ・ポインタ・カウンタ128の内容がCPU 12Aのものに従い、それもまた一サイクル遅延する、ということに注目する。それゆえに、CPU 12Aでのその到着よりも一サイクル後で記号AがCPU 12Bに到着しても、両方は、キュー126の“バイト0”位置に記憶される。これは、（1）FIFOs 102が（以下に説明する処理を）同期で動作すべく同期され、かつ（2）プッシュ・ポインタ・カウンタ128が記号のソース、即ち、ルータ14AからのT_{CLK}によって生成されたクロック信号によってクロックされ、そのクロック信号が記号によって経験されたものと同じ遅延640に出会うからである。他方、プル・ポインタ・カウンタ130は、それらがCPU 12の packets 受信機94によって生成された局所受信機クロック（Rcv CLK）によってクロックされるので、

互いに常にマッチする。更に、これらの局所受信機クロックは、動作のデュプレックス・モードであるときに周波数及び位相封じ込みされる；それらは、いかなる遅延も経験しない。

【0209】遅延640を見る別の方法は、ルータ14AとCPU 12Bの間の通信経路（バス32_y）におけるパイプラインの一部としてそれを考えることである。遅延640は、最大遅延が記号にその記号がキューからプルされる少なくとも1クロック・サイクル前に記憶キュー126をエンター（入力）させる限り、あらゆる値のものでありうる。CPU 12Aに伝達された記号は、実質的に、その複製がCPU 12Bのキュー126からプルされると同時にキュー126からプルされる前に1エキストラ・サイクル待つ。それは、ルータ14Aによって送信された記号ストリームの各記号が同時にCPU 12A，12Bのクロック同期FIFOs 102からプルされ、デュプレックス・モードで動作しているときに受信したデータの要求された同期を維持するというこの方法においてである。実質的に、CS FIFOs 102のキュー126の深さは、ルータ14AからCPU 12A，12Bへの二つの経路に同じ遅延を与えるべく調整する。表6を参照して説明された動作を達成するために、図45に示されたリセット及び同期処理が用いられる。処理は、デュプレックス・モード動作に対してCPU 12A，12Bのクロック同期FIFOs 102を初期化するだけでなく、デュプレックス動作に対してルータ14A，14Bの各々のCPUポートのクロック同期FIFOs 518（図27）を調整すべく動作もする。リセット及び同期処理は、SYNC CLK信号970（図46）によって表された、時期を始動し、CPU 12A及び12B並びにルータ14A，14Bの対応クロック同期FIFOsをリセットしかつ初期化すべくSYNC指令記号を用いる。（SYNC CLK信号は、システム10の素子、特にルータ14A，14B及びCPU 12A，12Bへの分配（分布）のためにクロック発生器654（図36）によって生起される。それは、クロック同期FIFOsにより記号を受信すべく用いられる、T_{CLK}よりも低い周波数のものである。例えば、T_{CLK}がおおよそ50MHzであると

ころでは、SYNC CLK信号は、おおよそ3.125MHzである。)

ここで図45を参照すると、リセット及び初期化処理は、それらが同じクロック信号から導出されるようにCPU 12A, 12B及びルータ14A, 14Bによって用いられたクロック信号を送信(T_CLK)及び装置の局所クロック(Local CLK)クロック信号としてスイッチすることによってステップ950で開始する。T_CLK及びLocal CLK信号は、実質的に同じ周波数であるが、種々のクロック信号を伝達することにおいて固有な遅延により同相である必要はない。更に、CPU 12A, 12Bにおける構成レジスタ(インターフェイス装置24の構成レジスタ74)及びルータ14A, 14Bにおける(ルータ14A, 14Bの制御論理回路装置509に含まれた)構成レジスタは、Frequency Lock状態にセットされる。

【0210】以下の説明は、ステップ952を含み、かつインターフェイス装置24(図9)、ルータ14A(図27)、図45及び図46を参照する。周波数封じ込み動作におけるクロックで、CPU 12Aは、SLEEP指令記号を送り始めることをそれに指令すべくオフ・ラインCPU 12Bにメッセージ・パケットを送る。次に、また、CPU 12Aは、ルータ14AにSLEEP指令記号を送り始め、さもなければ送られうるREADY指令記号を置換し、自己アドレス指定されたメッセージ・パケットが続く。SLEEP指令記号が受信されかつルータ14Aによって認識されたときに受信されかつ再送信される処理のメッセージ・パケットは、終了させられる。しかしながら、更なるメッセージ・パケットは、一つを除いて、近づけられない(離される(held off)): CPU 12Aからの自己アドレス指定されたメッセージ・パケット。それらのメッセージ・パケットは、受信され、かつ(宛先アドレス毎に)CPU 12Aへルータ14Aによって戻される。SLEEP指令記号は、同期処理に対してルータ14Aを“静止(quiet)”すべく動作する。CPU 12Aによって送られた自己アドレス指定されたメッセージ・パケットは、CPU 12Aによって受け戻されたときに、SLEEP指令記号の後に送られた自己アドレス指定されたメッセージ・パケットがルータ14Aによって最後に処理されなければならないので、ルータ14Aが静止状態であることをCPUに知らせる。

【0211】ステップ954では、CPU 12Aは、SLEEP指令記号を送ることの始動に続いてそれが送った自己アドレス指定されたメッセージ・パケットをそれが受け戻したかどうかを調べる。それがそのメッセージ・パケットの戻りを見て、かつそれによりルータ14Aが更なるメッセージ・パケットを一時的に処理しないことを確信したときに、CPU 12Aは、SYNC指令記号をルータ14Aに送るべくステップ956をエンター

(入力)する。そのSYNC指令記号がルータによって受信され、かつ指令復号論理回路544(図29)のようにもよって認識されたときに、制御論理回路509は、知らされる。制御論理回路509は、CPU 12A, 12BにエコーバックされるSYNC指令記号を生成すべく(ステップ958)、ポート出力5044, 5045の指令記号発生器632(図34)に信号を送るために、SYNC CLK 970の次の立上りエッジ(時間 t_1 - 図46)を待つ。次に、ステップ960(及びSYNC CLK 970の時間 t_2)で、ルータの制御論理回路509は、CPU 12A, 12Bから直接的に記号に受信するルータの入力論理回路5054, 5055に含まれる二つのクロック同期FIFOs 518に印加されるRESET信号972をアサートする。RESETは、アサートされる間、パワー・オン・リセット処理に関連して上述したように、互いに離れた所定数(この例では、2)離れた位置を記憶キュー518の位置で指し示すべく既知の状態にセットされたプッシュ及びプル・ポインタ・カウンタ530, 532(図29)で一時的非動作リセット状態に二つのクロック同期FIFOs 518を保持する。

【0212】同様に、SYNC記号は、ルータ14A, 14BによりCPU 12にエコーバックされる。CPU 12受信SYNC記号のそれぞれがパケット受信機96の記憶及び処理装置(図9及び図10)によって検出されると、RESET信号を各CPU 12のパケット受信機96(実際には、記憶及び処理素子110; 図10)によりアサートさせる。RESET信号は、CPU 12のクロック同期FIFOs 102(図10)に印加される。このCPU RESET信号は、リセット状態に両方のCPU 12のCPUクロック同期FIFOs 102を同様に保持し、それらの記憶キュー126(図11)、及びプッシュ及びプル・カウンタ128, 130を既知の状態に置く。ステップ962では、SYNC CLK 970信号の時間 t_3 で、CPU 12A, 12Bとルータ14A, 14Bとの間の記号送信を伴う送信機クロック信号(T_CLK)は、一時的に停止される。ステップ963(時間 t_4)で、CPU 12及びルータ14A, 14Bは、RESET信号をディ・アサートし、かつCPU 12A, 12B及びルータ14A, 14Bのクロック同期FIFOsは、それらのリセット状態から解放される。ステップ964(t_5)で、ルータ14A及びCPU 12は、T_CLKへの送信を再開しかつリンク上の最大予想遅延(maximum expected delay)に対する調整を許容するショート構成可能遅延(short configurable delay)を始める。遅延の終りで、ルータ14A及びCPU 12は、それらの対応クロック同期FIFOsからデータをプルすることを再開しかつ通常の動作を再開する。ルータ14Aのクロック同期FIFOsは、(IDLE記号にRESETによつ

て予めセットされた) キューから記号をプルすることを開始し、かつT_Clkは、キューに記号をプッシュすることを開始する。T_ClkでCPU12Aから受信した最初の記号は、例えば、付随T_Clk信号でキュー位置0 (またはプッシュ・ポインタ・カウンタがリセットされた値により指し示された他の位置) においてクロック同期FIFOにプッシュされるということに注目する。同様に、CPU12Bからの最初の記号は、FIFOキューの位置にそしてまた位置0 (またはRESETによりプッシュ・ポインタ・カウンタがセットされた他の位置) において配置される。ルータ14Aのクロック同期FIFOsは、ルータ14A及びCPU12A, 12Bの間で、他のものに関して、一つの通信経路に存在しうるいかなる遅延640を收容すべくここで同期される。

【0213】同様に、同じ仮想時間において、両方のCPU12A, 12Bのクロック同期FIFOs102の動作は、再開され、それらをルータ14Aに同期する。また、CPU12A, 12Bは、READY記号のためにSLEEP指令記号を送ることを中止して、適当なときに、メッセージ・パケット送信を再開する。それは、ルータ14Aに対する同期処理を終了する。しかしながら、処理は、ルータ14Bに対しても実行されなければならない。それゆえに、CPU12Aは、ステップ952に戻りかつ今回はルータ14Aの代わりにルータ14Bでステップ952-966を再び実行し、その後全てのCPU12A, 12B及びルータ14A, 14Bは、周波数封じ込みモードで動作すべく初期化される。デュプレックス・モード動作に対して残っているものは、二つのCPU12A, 12Bを同じ動作状態に置き、かつ本質的に同じ瞬間にそれらに同じ命令を実行する。再統一と呼ばれる、二つのCPUを同じ動作状態に置くことは、以下に説明する。しかしながら、まず、CPU12A, 12Bがデュプレックス・モードで動作していると想定して、デュプレックス動作からCPUの発散を結果として生ずる、可能な誤りを検出しかつ処理するために用いる方法及び装置を説明する。

【0214】**発散検出及び処理**：デュプレックス・モード動作は、CPUレベルでフェイル機能的フォルトトレラントを実施する。一対のデュプレックスされたCPU (例えば、システム10のCPU12A, 12B - 図1) の各々は、他の実質的に同一複製 (コピー) であり、状態及びメモリ内容を含み、かつ両方ともに、実質的に同時に、同一命令ストリームの同一命令を実行し、論理的、フォルトトレラントCPUを形成する。CPU12A, 12Bの一つまたは他のものの故障(failure)は、そのフォルト (故障) が検出されかつ適切に処理される限り - システム10の動作を停止、またはスローダウンすらしめない。故障しているCPUの検出は、デュプレックス・モード動作の明らかな結果 (帰

結) を用いる：両方のCPU12A, 12BのI/O出力は、適切なデュプレックス・モードに対して記号毎に同一である。それゆえに、適切な継続デュプレックス動作を確信するためになされることが必要な全てのことは、デュプレックスされたCPUのI/O出力を記号毎に比較することである。故障しているCPUは、他の動作の状態から発散し、結果的にその発散は、CPUのI/O出力にそれ自身を明らかにする。

【0215】図47は、ルータ14A, 14Bで発散を最初に検出するために用いる手順を示し (ステップ1000, 1002)、次いで、できるだけ早く故障しているCPUを終了すべく優雅なマナーでその発散を処理し、かつシステム10の残りに不良データを伝播することからそれを排除する。それゆえに、図47のステップ1000は、一つの論理CPUとしてロックステップ同期で動作するCPU12A, 12B (図1) のデュプレックスされたペアを有する。周期的に、CPU12Aは、サブプロセッサ・システム10A, 10Bの一つまたは他の周辺装置向けメッセージ・パケットを介してI/Oデータを送信する。出力メッセージ・パケットの宛先により、ステップ1002は、ルータ14Aまたは14Bの一つがそのI/Oデータを受信しているのを見て、それが受信されたならばCPU12Aからのメッセージ・パケットの各記号をCPU12Bからのものと比較する。比較は、通常設計の比較回路 (図示省略) によりCPU12A, 12BからI/Oを受信すべく接続されたポート入力5024及び5025の入力論理回路505の出力で行われる。受信した記号が同じであるならば、手順は、ステップ1000及び1002に残る - 適切な動作を示す。

【0216】比較ステップ1002が異なる記号を検出したならば、ルータ14の比較回路 (図示省略) は、ルータ制御論理回路509にERROR信号を発行し、発散を検出しているルータ14が両方のCPU12A, 12BにDVRG指令記号を送信するステップ1004に処理を移動させる。好ましくは、ルータは、発散を知らせることとどのCPUが継続するものであるかを知らせることとの間の時間を最小にすべくDVRG記号を送る前にできるだけ長く待つ。脇道にちょっとそれて、発散を検出するこの技術によって達成される求められた多数の対照的ゴールを説明することは、この地点で利点的でありうる：まず、ルータ14Aまたは14Bは、システムの残りへの誤りの伝播を防ぐために速やかな措置をとることが必要である。それゆえに、発散が検出されたにもかかわらず、ルータは、メッセージ・パケットの終了記号を除いて、その指定された経路にメッセージ・パケットを継続するかまたは送る：状態記号、“このパケットは、不良” (TPB) または“このパケットは、良好” (TPG) 状態記号。この記号なしで、ダウンストリーム (下流) 宛先は、受信したメッセージ・パケット

を用いない。

【0217】第2に、できるだけ少ないメッセージ・パケットが分断されなければならない。以下に更に説明するように、CPU s 12 A, 12 Bの一つは、“気に入った”または主CPUと指定され、かつCPU s がデュプレックス・モードで動作しているときに、ルータの気に入ったCPUからのメッセージ・トラフィックだけが送信される。分断は、もしあれば、どのCPUがフォルトでありうるかの、決定を行うことができるまで、発散を検出するにもかかわらず、ルータにメッセージ・パケットを送信することを終了させることによって最小にされる。気に入ったCPUでないならば、メッセージ・パケットは、終了記号の送信によって解放される。この場合TPG記号。第3に、発散を検出しているルータは、どの誤りが発散を生成するためにトランスパイアされた(transpired)かを正確に決定することが必要である。それは、これに簡単なリンク誤り、リンクレベル“キープアライブ”記号の損失、及びCRC誤りを探索させる。CPU s 12は、それらが、発生すべきリンクレベル・キープアライブ・タイム・アウトに対してDVRG記号を受信した後に十分な時間を許容する。(簡単なリンク誤りも検出することなく)発散を検出しているルータは、DVRG記号で発散を知らせる前にメッセージ・パケットの終りを待つことによって受信したメッセージ・パケットのCRCを検査する時間をそれ自身が買う。

【0218】最後に、かつ第4に、システム10は、TNetトランザクション・タイムアウトまたは支持不可能I/O遅延をもたらすことを回避するためにショート・バウンディド(short bounded)時期に発散処理を終了しなければならない。このゴールは、CPUからのメッセージ・パケットの結論を待つことがかなりの量の時間を費やすように、(終了状態記号の送信を見合わせることににより)メッセージ・パケットの解放の保持とある程度矛盾する。しかしながら、そのような遅延は、メッセージ・パケットを送信するためのCPUに対する最悪ケース時間(worstcase time)が保証されるならば、TNetタイムアウトをもたらすことができない。CPU s 12は、DVRG記号の受信により、それぞれが、その内でCPU s 12がそれらのいずれが故障したかを決定することを試み、かつ動作を終了し、かつそれらのどれが継続すべきかを決定することを試みる所定期間を制定するために用いられるタイマーをスタートする(ステップ1006)。更に、両方のCPU s 12 A, 12 Bは、両方のルータ14 A, 14 BにDVRG指令記号をエコーバックする。このエコーバックされたDVRG記号を受信する、ルータ14 A, 14 Bが発散を検出しないか、またはDVRG記号を予め見ていないならば、それは、DVRG記号をCPU s にもエコーバックする。この方法でDVRG指令記号をエコーバックすること

は、CPU s 12及びルータ14 A, 14 Bが全てDVRG記号を見ておりかつ可能な発散に気付いていることを確実にする。

【0219】一度、発散が検出され、CPU s (またはルータ)の一つの故障が示されることが全ての考慮したもの(CPU s 12 A, 12 B及びルータ14 A, 14 B)に対して明瞭であるならば、その故障のあらゆる結果が、不良データの形で、システム10の残りに伝播されないということを確実にするために配慮が払われなければならない。同時に、システム10は、フォルトのトレラントでありかつ走行し続けなければならない。それゆえに、(CPU s からの)出力パケット送信は、ルータが発散を検出と同時にCPU s からくるメッセージ・パケットが良好か不良かを決定することができるまで、少なくとも部分的に、続けなければならない。更に、発散をもたらした(divergence-causing)CPUは、決定され、かつ透過的に(即ち、外部干渉なしに)システムから除去されなければならない。この後者のタスクは、CPU s 12の責任であり、一度発散及びあらゆる誤りがCPU s 12に知らされたならば、それらは、それらのどれが動作を続け、どれが更なる動作を終了しかつそれによってシステム10からそれ自身を効果的に除去するかそれら自身の間で決定しなければならない。

【0220】それゆえに、発散ルーチンのステップ1006は、CPU s 12 A, 12 Bの各々にそれらに供給された種々の誤り表示を解析させる；この誤り解析は、以下に更に説明する。しかしながら、ちょっとした間、不良データの伝播を制限すべく発散を検出したルータ14の機能は、説明を必要とする。DVRG記号がルータ14から発行されたか、または受信された後、CPU s から受信しかつ発散が検出されたか、またはDVRG記号が受信されたときに送られる処理で受信した全ての更なるメッセージ・パケットは、パケットを終了する状態記号を除き、ルータを通過される；即ち、TPG(このパケットは、良好)またはTPB(このパケットは、不良)状態インジケータ記号。デュプレックス動作の間中、上記で簡単に説明したように、ルータ14 A, 14 Bの各々は、制御論理回路509(図27)に含まれる構成レジスタ(図示省略)にセットされたビット位置の側に“気に入った”CPUを有すべく構成される。デュプレックス動作では、ルータは、の気に入ったCPUから受信したメッセージ・パケットを再送信する；他のまたは“気に入らない”CPUからのメッセージ・パケットは、発散検出に対してだけ用いられる。ルータは、TPG/TPB状態インジケータ記号を後ろに追加することによりパケットを“解放する”前に(ステップ1014)、それらのどれが動作を継続するか、どのルータ14 A, 14 Bが知らされるか(ステップ1012)というCPU s によってなされた決定を待たなければならない。ルータが気に入ったCPU 12が継続すべく決定さ

れたものであることを知らされたときに、ルータは、TPG状態インジケータ記号を追加しかつ送ることによってメッセージ・パケットを解放する。逆に、ルータが別を、即ち、それが継続する気に入ったCPUでないことを、知らされたならば、メッセージ・パケットは、TPB記号を追加することによって廃棄される。

【0221】損失するデータの量を制限するために（上記第2のゴール）、二つのルータは、ことなる好み(favorites)で構成される（例えば、ルータ14Aの好みはCPU12Aであり、ルータ14Bの好みはCPU12Bである）。続けて、一度、検出された発散がCPU12A、12B及びルータ14A、14Bにブロードキャスト（放送）されたならば（ステップ1004）、CPU12A、12Bの各々は、発散のフォルトがどこにあるかをそれぞれ個別に決定しようとしてステップ1006で状況（状態）を評価し始める。一度、CPU12A、12Bのどれが故障したかが決定されたならば（ステップ1008）、そのCPUは、それ自身の動作を終了し（ステップ1012）、シンプレックス・モードにもかかわらず、動作を継続すべく他のものを残す。CPU12A、12Bがそれらのどれがフォルトでありうるかを検出されたかまたは報告された誤りから決定することができない場合には、それらは、各CPUのインターフェイス装置24の構成レジスタ74（図9）の一つに含まれる“タイ・ブレーカー”ビットに頼る（ステップ1010）。ステップ1006にちょっと戻ると、CPU12A、12Bの故障しているものがどれであるかの決定は、CPU12A、12Bとルータ14A、14Bをリンクしている通信経路上にどの誤りが検出されたかに主に基づく。ルータ14A、14Bが発

散に気付いた後、それらの各々は、上述したように、通常の動作を継続する：発散を示している記号差が検出されたか、またはその後受信されたときにCPU12A、12Bからルータ14A、14Bによって受信した単一メッセージ・パケットは、終結状態インジケータ記号を除きルータを通過される。両方のルータ14A、14Bは、例えば、検出されたCRC誤り、指令記号誤り、またはリンク誤りの他の型のような、可能リンク・プロトコル誤り及び／又は違反に対して特定のルータ14をCPU12に相互接続している通信経路（TNetリンクL）を監視し続ける。そのような誤り／違反がリンクL（例えば、リンクLx - 図1）上で検出されたならば、検出ルータ14A、14Bは、This Link Bad（このリンクは不良）（TLB）記号をそのリンク（Lx）上のCPUに送り戻す。同時に、検出ルータは、Other Link Bad（他のリンクは不良）（OLB）記号を（他のリンク、Ly上の）他のCPU12にも送り戻す。それにより、両方のCPU12は、ルータ14A、14Bの一つへの通信経路（リンク接続Lx）、またはそのリンクに関連した何かが故障でありかつ信頼できないという事実を気付かされる。それは、どのCPUが動作を終了すべきかを決定することを試みるために、ステップ1006で、CPUが用いる情報であり、シンプレックス・モードにもかかわらず、他のCPUを機能し続けさせる。CPUは、この誤り情報を累積し、以下の表7に示したものに類似する表を生起する。

【0222】

【表7】

表 7		局所CPUのアクション
局所ルータ	遠隔ルータ	
(1) 誤りなし	誤りなし	主に継続；代替は終了
(2) 誤りなし	TLB又は誤り	終了
(3) 誤りなし	OLB	継続；他のCPUは終了
(4) TLB又は誤り	あらゆる状態	終了
(5) OLB	あらゆる状態	継続；他のCPUは終了

【0223】表7は、CPUによって検出されうるかまたはそれに報告されうるし、かつそれによりCPUが発散を知らされたにもかかわらず何をするかを決定しうる5つの可能条件を示す。表7に用いたように、“局所”は、CPUとして同じサブプロセッサ・システム10A、10Bに含まれたルータ14A、14Bを称することを意味する。例えば、図1を参照すると、ルータ14Aは、CPU12Aに対して“局所”であるが、CPU12Bに対して“遠隔”である。表7に用いたように、“Primary（主）”は、上記したタイブレーカー・ビットを称する：各CPUのインターフェイス装置24（図9）の構成レジスタ74の一つに含まれるビット。第1の状態にセットされたときに、その特定CPU

12は、それにより、主（1次：primary）として識別され、逆に他のCPU12は、それが“代替”であることを示すべく反対の状態にセットされたその同じ構成ビットを有さなければならない。これらのビットは、CPUの別のものがその決定をすることができないときに継続すべくCPUの一つを選択するために発散処理の間に用いられる。局所ルータが誤りなしを報告する全ての場合に、CPUは、決定を行うことを遅延することに注目する。これは、他のCPUが誤りを検出しかつ自己検査しえたし、かつ局所ルータがキープアライブ記号の損失を実質的に検出し、かつOLB記号により局所CPUに誤りを報告するという可能性を許容する。

【0224】“Any Status (あらゆる状態)”は、それをちょうどを称する：遠隔ルータからの報告のいかんを問わず（誤りの表示、または誤りの無表示）、局所CPUは、見出し“Action of Local CPU (局所CPUのアクション)”の下に示されたアクションを行う。“Action of Local CPU”は、CPU s 12 A, 12 Bの特定のものによって取られたアクションを表わし、表の行の一つに示されたようにその特定CPUによって見られた条件を与える。例えば、行4にセットされた条件がCPU 12 A (ルータ14 Aが誤りを報告したか、またはCPU 12 Aが誤りを検出した)によって見られたならば、CPU 12 Aは、それがデュプレックスされたペアの他のもの、CPU 12 B、に動作を継続させるために、動作を終了すべきという決定を行う。逆に、行4の条件は、他のCPU 12 Bは、OLB記号をその“遠隔”ルータから受信することを示し、ルータ14 AとCPU 12 Aとの間の通信経路が疑われるという事実を知らせている。CPU 12 Bの視点から、これは、行3または5によって表された条件でなければならない。ルータ14 A, 14 Bの一つだけ（この場合には、ルータ14 A）が誤りを検出したならば、行3表示がCPU 12 Bに与えられる。両方のルータ14 A, 14 Bが誤りを検出した場合には、それぞれは、TLBをCPU 12 AにかつOLBをCPU 12 Bに知らせる。CPU 12 Bは、ルータ14 BからOLBを見て、これを行5条件にマッチし、IOY記号をルータ14 Bに発行して、継続する。

【0225】表7の行4及び5は、ある問題を許容することに注目する。例えば、ルータ14 AがTLBをCPU 12 Aに知らせかつルータ14 BがTLBをCPU 12 Bに知らせたならば、両方のCPUは、それら自身を殺す（凍結する）。しかしながら、両方の局所ルータが誤りを生起することが起こりえない場合のような、変わった想定ではない、ほとんどの場合一つのフォルトだけが所与の時間に発生すると想定したならば、表7の条件は、受容可能である。さもなければ、ルータ14とCPU 12との間の二つ以上のリンク上の多重誤りが発生するならば、システムは、生き残る必要がない。同様に、両方のルータがOLBsをそれらの局所CPUsに知らせたならば、両方のCPUsは、取って代わることを試みるであろう。これは、クロック故障を表わす傾向がある。クロック回路は、そのような誤りを検出しかつ故障しているCPUを凍結すべきである。従って、表7は、CPU s 12及びルータ14 A, 14 Bが検出できる誤り表示を表わす。一般に、CPU 12がその局所ルータから誤り表示を受信したならば、それは、ペアの他のものを継続させるために動作を終了する。主/代替指定に頼ることは、いずれのCPUも、(DVRG指令記号の受信により始動された)各CPUのタイマーの期間

満了においていかなる種類（行1、表7）の誤り表示を受信しないときにだけ生じる。この場合には、各CPUの主構成ビットに頼ることによりタイ(tie)がブレイクされる。主として識別されたものは、継続しかつ他のものが終了されたと想定する；それ自身の構成ビットにより代替と識別されたCPUは、その動作を終了する。

【0226】それゆえに、CPU s 12は、どれが継続し、かつどれが継続しないかの決定を行い（ステップ1008）、次いで、ステップ1006, 1010の一つにおいてなされた決定により一つのCPUが終了するステップ1012に続く。終了するCPU 12は、自己検査及び凍結を誘導することによってそのようにする。継続しているCPUは、ルータが継続シテイルCPUだけを頼り、他のCPUからの全ての送信を無視すべきであるということを示すべくIOY記号(I Own You)をルータ14 A, 14 Bに送る。それに応じて、制御及び状態装置509 (図27)内の状態マシン(図示省略)は、上述した“気に入った”ビットを変える。2～3の例は、発散の概念を理解することを容易にする。図1を再び参照して、CPU s 12 A, 12 Bがデュプレックス動作モードで動作し、CPU 12 Aが全ての後続I/O動作がCPU 12 Bのものとは異なるようにフォルトを被ることを想定する。従って、次のI/O送信で、ルータ14 A, 14 Bの一つ（データが指向されるもの；または故障がCPU 12 Bのものとは異なる宛先にI/Oを指向しているCPU 12 Aを有するならば両方）は、発散を検出する。しかしながら、それらは、上述したように、パケット全体が現在のメッセージ・パケットCRC検査が通るかどうかを決定すべく受信されるまで、または簡単な誤りに出会うまで、待ち、その時に各ルータが両方のリンクL上のDVRG記号を送信する。両方のルータはプロトコル誤りを見る（理解する）と想定する。検出されたプロトコル誤りは、両方のCPU s 12にDVRG記号を送っているルータ14 A, 14 Bをすぐに結果として生じ、誤りが検出されたことによりリンクL上にThis Link Bad (TLB)記号を送り返す、即ち、リンクLx, Lyは、ルータ14 A, 14 Bを、それぞれ、CPU 12 Aに接続している。それらがTLB記号を送ると同時に、両方のルータ14 A, 14 Bは、Other Link Bad (OLB)記号をCPU 12 Bに送る。CPU 12 Aは、DVRG記号の受信により、その記号をルータ14 A, 14 Bにエコーバックし、その内部発散処理タイマーをスタートし、継続すべきか終了すべきかの決定を始める。その局所ルータ14 AからTLBを受信して、CPU 12 Aは、それがCPU 12 Bの継続を許容するために終了しなければならないということを示すべく決定する（行4、表7）。

【0227】更に、このシナリオにおいて、CPU 12 Bは、両方のルータ14 A, 14 BからOLB記号を受

信しかつそれが継続すべきCPUであることを報告するそれらから決定する。従って、IOY記号を両方のルータ14A, 14Bに発行する。それに応じて、ルータ14A, 14Bは、CPU12Bによるパケット送信だけが実行され、かつCPU12Aからの送信が無視されるようにそれら自身を構成する。また、発散検出は、故障しているルータを検出する。例えば、ルータ14Aは、それに発散作用を実行させるような方法で故障し、二つのCPU12A, 12Bを発散されると想定する。良好なルータ、ルータ14Bは、この発散を検出し、かつCPUのそれぞれにDVRG記号でそれを知らせる。各CPUは、両方のルータ14A, 14BにDVRG記号をエコーバックする。特定のルータ14Aの故障により、それは、DVRG記号をCPUにエコーバックしうるしまたはしえない。CPU12Aは、それがそれを故障したルータ14Aに接続しているその局所リンクに誤りを有することを見出し、それゆえにそれは、それが自己検査しかつ終了しなければならないことを決定する。逆に、ルータ14Bは、この終了を検出し、かつTLB記号をCPU12Aに、かつOLB記号をCPU12Bに戻す。次いで、CPU12Bは、両方のルータにIOY指令記号を発行する。

【0228】上述したのは、ルータ14A, 14Bの一つまたは他のものまたは両方が発散を検出し、CPUにDVRG記号を発行するが、CPU12A, 12Bまたはルータ14A, 14Bのどちらも誤りを検出しないような“クリーン”発散であった。従って、“主”CPUは、初期化の間中に構成レジスタに予めセットされたように、それが継続しかつ両方のルータ14A, 14BにIOY記号を発行しなければならないことを決定する。同時に、“代替”CPU12Bは、自己検査し、かつ終了する。上述したものに加えて、発散をもたらすことができる誤りまたはフォルトの型は、以下のものを含む：

— 訂正不可能なメモリ誤り、誤りの可能な伝播を排除するためにCPUに動作をすぐに凍結させる。CPUは、ルータ14A, 14Bに対して死んだように見え、それらにTLB記号を機能CPUへ、かつOLBを他の（稼動している）CPUへ送らせる。稼動しているCPUは、それが継続することを決定し、かつIOY記号を両方のルータ14A, 14Bへ送る。

— 報告された誤りなしでCPU12を発散させるソフトウェア欠陥。これは、（プロセッサ20上で走っている）ソフトウェアが、状態を変更すべく既知の発散データを用いるときのみ発生することができる。例えば、各CPU12が異なる通し番号を有するものと想定する（例えば、アドレス空間の読取り専用またはプログラマブル読取り専用領域に保持される）。CPU12Aの通し番号は、CPU12Bのものとは異なる。プロセッサが、（例えば、通し番号がある値の後にくるならば

ブランチングにより）実行される命令のシーケンス（順番）を変えるために、またはプロセッサ・レジスタに含まれる値を変更するために通し番号を用いるならば、CPU12の完全な“状態”は、異なる。そのような場合には、“非対称変数”機構（例えば、ソフト・ヴォート機構）が用いられる。これは、それらが、同一の決定を行いかつ同一のデータに基づき同一の動作を実行することができるように、二つのCPU12に情報を交換させる。この例では、CPUは、両方がコピーを有するようにそれらの通し番号の値を交換する。次いで、CPU12Aの通し番号がある値の後にきたならば、それらは、コード（符号）を実行し、このシーケンスを実行する、等。主CPUの所定の構成、及び最終選択は、単に一つのCPUに、及びそれによりシステム10に、ソフトウェア介入なしで、処理を継続させる。

【0229】— CPU12のインターフェイス装置24の出力における誤りは、誤りが発散として発生するリンクLにより、ルータ14A, 14Bによって検出され、かつTLB/OLB状態は、CPUに戻され、DVRG記号が後に続く。CRC誤り、及び指令記号汚染は、同様に検出及び処理される。

— CPU12に接続するルータ14A, 14Bの出力上の誤りは、影響を及ぼされたCPUのインターフェイス装置24によって検出される。CPUは、故障しているルータ14にTLB記号を送り、図45及び図46の発散処理ルーチンを起動すべくそのルータによる発散として検出される。誤りを検出しているCPUは、終了し、かつ継続しているCPUは、両方のルータにIOY記号を送る。発散は、システム10（図1）に、デブプレックス・モード動作で動作しているときに、可能な故障を検出させ、かつ外部介入なしで、かつシステム・ユーザに透過的に、ユーザ・オペレーションを停止させることなくシステムへの誤りの伝播を未然に防ぐかまたは制限すべくシステムから故障している装置（CPU12Aまたは12B、またはルータ14A, 14B）を除去させる。故障したのがCPU12であったならば、CPUは、回線からはずされ、かつ検査済みの、機能するCPUで置き換えることができる。デブプレックス動作において、その代替装置(replacement)をオン・ラインに戻すことが次のセクションの主題である：再統一。

【0230】再統一：

— 概要

このセクションは、“再統一(reintegration)”を通して二つのCPU12をロックステップ、デブプレックス・モード動作に至らせるために用いる手順を説明する。説明は、図1に示された処理システム10の部分を形成しているCPU12A, 12B、ルータ14A, 14B、及び保守プロセッサ18A, 18Bを参照する。更に、説明は、それらが機能する方法であるので、単一装置としてプロセッサ20a, 20b、インターフ

エイイス装置24a, 24b、及びCPU s 12A, 12Bのメモリ・コントローラ26a, 26b (図4)を参照する。再統一は、最初にオンラインにもってこられたか、またはある時間シンプレックス・モードで動作した後、またはシステム10の先のデュプレックス・モード動作が発散を結果として生じ、かつ故障している素子

(例えば、CPU s の一つ) が除去されかつ置換された後に、デュプレックス・モード動作に二つのCPU s を置くために用いられる。

【0231】再統一は、まだ動作している(即ち、オン・ライン状態の)CPU s 12の一つで始めなければならない、外部介入なしで、再統一がバックグラウンドで実行され、従ってユーザに対して実質的に透過であるので、かなり確実にユーザ・アプリケーションを実行する。他のCPU 12は、それがユーザ・コードを走らせないという意味で、オフ・ライン状態である；それは、それにその初期化及び再統一に要求される最低限のタスクを実行させるべく十分なコードを走らせる。この初期化は、それらが、事実上同時に同一命令ストリームの同じ命令を実行することができるようにデュプレックス・モード動作に対して事実上同じ状態に二つのCPU s 12を配置することを含み、結果として同じアクションをとる。また、再統一は、発散検出を実施することができ、かつCPU s 12向けのメッセージ・トラフィックが事実上同時にペアになったCPU s の両方に送付されるように、ルータ14A, 14Bがデュプレックス・モード動作に対して構成されることを結果として生ずる。図48～図51のフロー図によってある程度より詳細に概説された、一つのオン・ラインCPUのシンプレックス・モード動作から二つのCPU s のデュプレックス・モード動作に変更するための処理における主要ステップは、一般に次の通りである：

1. 遅延した(“シャドウ”)周波数封じ込み、デュプレックス・モード動作に、二つのCPU s (一つがオン・ライン、他のものがオフ・ライン)及びそれらの接続ルータをセットアップしかつ同期し、別個の(distinct)命令ストリームを実行する；
2. オン・ラインCPUのメモリをオフ・ラインにコピーし、実行されておらず、かつオフ・ラインCPUに、コピーされることが必要でありうる、オン・ラインCPUのメモリにおける変化を監視するトラッキング処理を維持する；
3. 同じ命令ストリームからの遅延した(スレーブ)デュプレックス・モードを走らせるべくCPU s をセットアップしかつ同期する(ロッーステップ動作)；
4. オン・ラインCPUからオフ・ラインCPUに全ての残っているメモリ位置をコピーする(このステップは、全てのメモリが読取られるまでオフ・ライン・メモリの各位置を読取り、かつオフ・ラインCPUのそれらと異なると疑われるそれらのメモリ位置だけをコピーす

る)；そして

5. 二つのCPU s の完全ロッーステップ、デュプレックス動作を起動する。

【0232】— セットアップ

ここで図48を参照すると、再統一手順が入力される前に、CPU s 12A, 12B及びそれらの最初の回線ルータ(即ち、CPU s に直接接続するもの)14A, 14Bがセットアップされなければならない。これは、MP18Aの使用を含む。ステップ1050では、MP18Aは、CPU s 12A及び12Bのインターフェイス装置24における制御レジスタ74の所定のレジスタ(図示省略)を、両方のCPU s が周波数封じ込みモードであるが、一つ(オフ・ラインCPU)は、遅延したまたは“シャドウ”ファッショで動作し、他のものに遅れて多数(例えば、8)のクロック・サイクルを動作するような(ソフトウェア動作の後)次の状態に書込む。CPU s 及びルータのこの動作のモードは、以後“シャドウ・モード(shadow mode)”と呼ばれる。ルータの構成レジスタ(図示省略)も、ステップ1052でMP18Aによって同様にセットされる。更に、構成レジスタは、“気に入った”としてルータ14A, 14Bへオフ・ラインCPU 12Aを識別するためにMP18Aによって書込まれる。これは、ルータ14A, 14Bに、シャドウ・モードにおけるときに送信に対してCPU 12Aだけを頼らせて、オフ・ライン12Bから出する全ての送信を無視する。

【0233】次に、上述したのと同じようなファッショでCPU s 12A, 12Bのクロック同期FIFOsとルータ14A, 14Bを同期するシーケンスが入力され(ステップ1060-1070)、次いでそれらをシャドウ・モード動作に移動する。シャドウ・モード動作は、オフ・ラインCPU 12Bに送られた送信が記号毎にオン・ラインCPU 12Aに送られたものに多数(例えば、8)のT_{clk}クロック遅れるということを除き、二つのCPU s 12A, 12Bが真性デュプレックス・モード動作で機能しているときと同じようなファッショで、ルータ14A, 14Bから同じメッセージ・パケット及び他のTNet記号を受信する ようなものである。即ち、ルータ14A, 14Bの一つから送信される記号は、その同じ記号がオフ・ラインCPU 12Bによって受信される8T_{clk}クロック前にオン・ラインCPU 12Aによって受信される。ステップ1060及び1062は、クロック同期FIFOsを同期するために図45、46の説明に関連して上述したのと同じステップを基本的に実行する。オフ・ラインCPU 12Aは、SYNC CLK信号で、CPU s 及びルータを同期すべく動作する、SLEEP記号、自己アドレス指定されたメッセージ・パケット、及びSYNC記号のシーケンスを送る。一度そのように同期されると、オフ・ラインCPU 12Aは、次に、ステップ1066で、ス

トップ1052でMP18Aによってセットされた次の状態にルータをまず移動すべく動作する、Soft Reset (SRST) 指令記号を送る。ルータ14A, 14Bは、いま、オン・ラインCPU12Aへ送られた全てのトラフィックが複製されかつオフ・ラインCPU12Bへも送られるが、オン・ラインCPU12Aへ送られたものに8クロック遅れるように送られるような、遅延したデュプレックス・モードである。更に、ルータ14A, 14Bは、メッセージ・パケットを受信するのにオン・ラインCPU12Aだけを頼るべくセットされ、かつオフ・ラインCPU12Bからのあらゆる送信を廃棄する。

【0234】SRST記号は、CPU12A, 12Bにエコーバックされる(オフ・ラインCPU12BへのSRST記号が8クロック遅れで)。CPU12A, 12Bによる受信により、CPUは、MP18Aによってセットされた次の状態動作モードへ移動される：シャドウ・モード。手順は、いま、メモリがオン・ラインCPU12Aからオフ・ラインCPU12Bにコピーされている間に行われるメモリ及び状態(例えば、レジスタ、キャッシュ、等)の監視をセットアップすべくステップ1080(図49)へ移動する。オン・ラインCPUの状態をオフ・ラインCPUにコピーするステップは、オン・ラインCPUの全ての進行中の動作を停止し、全ての構成レジスタ及び制御レジスタ(例えば、インターフェイス装置24の構成レジスタ74)、キャッシュ等をオン・ラインCPUのメモリ28に書き込み、メモリ28の全内容をオフ・ラインCPUにコピーし、かつそれらを一緒に持ち出すリセット・ルーチンに両方のCPUを移動する(vectoring) ことによって単に達成される。しかしながら、大きなシステムに対して、これは、達成するために数十秒またはそれ以上を費やするし、再統一のためにシステム10をオフ・ラインするのに受容できない時間の量である。この理由により、再統一処理は、オフ・ラインCPUに状態をコピーする動作のほとんどがバックグラウンドで行われる間にオフ・ラインCPUにユーザ・アプリケーション・コードを実行することを継続させる方法で実行される。

【0235】しかしながら、オン・ラインCPUがユーザ・アプリケーション・コードを実行し続けるので、どちらかと言えば、オフ・ラインCPUへの状態のコピーの間に、オン・ラインCPUのメモリ28のセクションは、コピーされた後で変更され、それゆえにコピー・メモリの最初のパスの結果で、オン・ラインCPUメモリは、オフ・ラインCPUのそれに一致しない。これに対する理由は、オン・ラインCPUのプロセッサ20による通常の処理は、それがオフ・ラインCPUにコピーされた後でメモリ内容を変更することができることである。(オン・ラインCPU12Aのメモリ28へのI/O書き込みは、それらがオフ・ラインCPU12Bに対し

てもなされるのでCPU12A, 12Bのメモリの内容を一致させないようにすべくコピー手順に影響を及ぼさない。) 再統一の間にこの問題を処理するために二つの機構が用いられる： まず、再統一処理の間中にメモリ位置がオン・ラインCPU12Aに書込まれたとき、それは、“汚れた(dirty)”とマークされる；第2に、オフ・ラインCPUへのメモリの全てのコピーは、コピーが確認される前に書込まれる(理解されるように、コピーされたメモリ位置を上書きする)ことからオン・ライン・メモリのデータを保護すべくコピーされるメモリ位置を監視する“アトミック・ライト(AtomicWrite)”機構によって実行される。両方の機構は、再統一手順の間だけ用いられ、かつ両方の理解は、メモリ・プリコピー及びオン・ラインCPUからオフ・ラインCPUへの状態の後続コピーの適切な理解を容易にする。

【0236】— メモリ・マーキング

メモリを“汚れた”とマークすることは、特にこの目的のためにECC検査ビットの一つの使用を含む。64ビット・データ語は、単一ビット誤り訂正、各記憶された語に対する二重ビット誤り検出を供給すべくECCコードの8ビットで記憶されるということが思い出される。ECCの8ビットの一つの使用は、1ビット訂正能力(機能)に影響を与えない。しかしながら、それは、2ビット誤りを検出するための能力(機能)を制限しうる。しかし、メモリ複写処理が比較的短い期間の間だけ続くので、このリスクは、受容可能であると信じられる。再統一の間中に、オン・ラインCPU12Aによるメモリ位置への各書き込み動作は、その同じメモリ位置の後の(later)読取り動作に位置を汚れたとマークするように解釈される所定のECCシンδροームを導き出さすべくECC検査コードの1ビットを反転する。(例外がある：オフ・ライン・メモリも同じI/Oデータで書き込まれるので、第1は、入力I/Oの書き込みである。第2は、アトミック・ライト(AtomicWrite) 機構に関連した書き込み動作(“書き込み条件付き(Write Conditional)”動作(operation)である。)この方法で、オフ・ラインCPU12Bのメモリ28によって見られえないオン・ラインCPU12Aのメモリ28の内容における変化は、マークされ、かつそれらの位置をオフ・ラインCPU12Bのメモリ28に続いてコピーすることによって“クリーン(clean)”にされる。

【0237】ECCのどのビットがマーキングに用いられるかは、どのビットが用いられても一致して(矛盾しないで)もちいられるならば、事実まったくそんなに関係ない。

【0238】— アトミック・ライト機構

オフ・ラインCPUへのオン・ラインCPU12Aの状態のコピーは、ルータ14A, 14Bの一つを介してTNet構造を用いているメッセージ・パケット送信の使用を含む。しかしながら、オン・ラインCPU12Aの

メモリ28へのI/O書込みがオフ・ラインCPU12Bのメモリへもなされるようにルータ14A、14Bがデュプレックス・モード動作にセットされるので、オン・ライン状態を含んでいるメッセージ・パケットの転送は、両方のCPU12A、12Bへ同様に転送されるが、良好な使用にする：オン・ラインCPU12Aによる状態包含メッセージ・パケットの受信は、ルータ14(14Aまたは14Bのどちらか用いられたもの)によるその誤りなし(error-free)受信及び再送信を確認する。また、それは、メモリ位置をクリーンにマークするAtomicWrite(アトミック・ライト)の書込み動作である。それゆえに、コピーされたメモリ位置は、それらが含むデータが、それらをオフ・ラインCPUへ転送し、かつそれが来たメモリへ書込まれる(戻される)メッセージ・パケットに戻し受信されるまでクリーンとマークされない。この方法で、位置は、オフ・ラインCPUのメモリへ成功裏にコピーされるという確認が行われる。

【0239】しかしながら、オフ・ラインCPU12Bへのコピーのためのオン・ラインCPU12Aのメモリ位置の読取りと、オン・ライン・メモリへコピーされたデータの後続の書戻しとの間で、CPU12Aのメモリ位置への別の書込み動作が行われると想定する(入力I/O書込み、またはプロセッサ書込み動作のいずれか)。(オフ・ラインCPUへコピーされた)戻りデータは、それが最初にきた位置へ書込まれ、位置が含まれている新しい値がなんであろうと上書きし、位置を良好とマークし、かつオン・ラインCPU12Aの進行中の動作に必要でありうるデータを破壊する。この問題を未然に防ぐために、AtomicWrite機構がファクションされた。AtomicWrite機構は、オフ・ラインCPU12Bへコピーされる処理においてメモリ位置を監視するためにオン・ラインCPUのメモリ・コントローラ26(図4)を用いる。メモリ・コントローラは、オフ・ラインCPU12Bへコピーするために読取られたが、それらをクリーンとマークすべく戻りデータでまだ書込まれていない、それらのメモリ位置のアドレスを、それらのレジスタにおいて、追跡すべく再統一の間に動作される一組のレジスタ及び制御状態マシンを用いる。コピーされたデータのCPUへの戻りの前にリスト上の位置への

(戻されたもの以外の)データの介入書込みが存在すると、リストは、それによってマークされる。コピーされたデータがオン・ラインCPUへ戻されるときには、それがメモリに書込まれる前にリストは、検査される。位置が合間(インターリム)に書込まれたとマークされたならば、戻りデータは、廃棄され、かつメモリ位置が汚れたとマークされたまま残される。他方、オフ・ラインCPU12Bへのコピーのために読取られているのでメモリ位置が書込まれていないならば、戻りコピー済みデータは、位置へ書戻されかつそれらの位置がクリーンとマークされる。

【0240】本質的に、AtomicWrite機構は、二つの動作を用いる：“ReadLinked(読取りリンクされた)”メモリ動作及び“WriteConditional(書込み条件付き)”動作。ReadLinkedメモリ動作は、オフ・ラインCPU12Bへコピーされるべきメモリ28におけるオン・ラインCPU12Aの位置を読取り、MC26によって保守されるリンク表におけるその動作のアドレスを保管し、かつそれがオフ・ライン(並びにオン・ライン)CPUへメッセージ・パケットとしてアセンブリされかつ送られるBTE88のキューにコピーされるべきデータを導入すべく機能する。ReadLinked動作のアドレスを保管することは、データをメモリ位置へ戻しかつリンク表のエントリをクリアする、将来のWriteConditional動作にそれを“links(リンクする)”。一般に、動作は、通常のブロック読取りであり、多数のメモリ位置からデータのブロックを生成する。リンク表に書込まれるアドレスは、メモリ位置のブロックのヘッドまたはエンドにおけるメモリ位置のものである。コピーされたメモリ位置のブロックから読取られたデータを含んでいるメッセージ・パケットがオン・ラインCPU12Aによって戻し受信されたときに、それは、WriteConditional動作でメモリ28へ書込まれる。しかしながら、データが書戻される前に、MC26は、リンク表を検査する。ブロック内のメモリ位置が別の動作によって書込まれたならば(例えば、プロセッサ20による書込み、I/O書込み、等)、その以前の書込み動作は、リンク表における位置にフラグを立てる(並びに書込まれたメモリ位置を汚れたとマークする)。MC26は、フラグに気付く、かつそれを書込むことなくWriteConditionalデータを廃棄し、汚れたとマークされたメモリ位置をそのまま残し、それらがまだオフ・ラインCPU12Bへコピーされなければならないことを示す。

【0241】再統一処理、及びここで図49に戻ると、メモリ追跡(AtomicWrite機構及びメモリ位置をマークすべくECCを用いて)は、ステップ1080及び1082でイネーブルされる。これは、再統一(REINT)信号をアサートさせるために再統一レジスタ(図示省略：インターフェイス装置24の構成レジスタ74の一つ—図9)を書込むことを必要とする。REINT信号は、WriteConditionals以外の全ての書込み動作、及び全てのI/O書込み動作、に対してECC論理回路85によって生成されたECCの8ビットの一つを反転するために各メモリ・インターフェイス70(図15)のECC論理回路85に結合され、それゆえに、連続的に読取るときに、この反転されたビットを有するデータは、汚れたとマークされているメモリ位置を識別するシンδροームを生成する。そのようにイネーブルされたメモリ追跡で、再統一手順は、オン・ライン・メモリの内容が、下から上へ(または、もし所望ならば上から下へ)、最初のパスでオフ・ラインCPU12Bのメモ

リヘコピーされる（ステップ1084）ような、“プリコピー（pre-copy）”シーケンス（ステップ1084-1088）へ移る。入力I/O及びAtomicWrite機構以外の書込み動作によって後で書込まれたメモリ位置は、書込まれた位置を汚れたとマークするためにECCビットを用いる。また、ReadLinked動作によってコピーされた後だが、後続のWriteConditional動作の前の、位置へのメモリ書込みも、マークされる。

【0242】メモリ28の全内容が一度走らされかつオフ・ライン・メモリヘコピーされた後、シーケンスは、オフ・ライン・メモリのそれに、即ち、ステップ1084の結果で汚れたとマークされたままのメモリ位置に一致しないかもしれないオン・ライン・メモリ位置の増分コピーをここで実行すべくステップ1086及び1088へ移る。増分コピーは、オン・ライン・メモリ全体を通る多数のパスを含み、合成シンドロームを検査するために各位置を読取る：位置がそれにより汚れたかまたはクリーンかマークされる。汚れたとマークされたならば、位置は、オフ・ラインCPUヘコピーされ、かつクリーンとマークされる。位置がクリーンとマークされたならば、そのまま残される。増分コピー動作全体を通して、オン・ライン・プロセッサの通常アクションは、あるメモリ位置を汚れたとマークする。増分コピーの多数のパスは、汚れたメモリ位置がコピーされクリーンにされる割合（速度）がメモリが汚される割合（速度）と実質的に等しい点に達するまでステップ2052で終了されることが必要である。これを行うために、カウンタは、ReadLinked、WriteConditional、故障したReadLinked、及び故障したWriteConditional動作に対するMC26に含まれる。メモリを通る各パスの終りでの成功したWriteConditional動作の数を注目することにより、プロセッサ20は、先のパスと比較した所与のパスの影響を決定することができる。利点が降下したときに、プロセッサ20は、プリコピー動作を諦める。この地点で再統一処理は、二つのCPU s 12A, 12Bをロックステップ動作に置くために準備する。

【0243】それゆえに、再統一手順は、ステップ1100でオン・ラインCPU 12Aがフォアグラウンド処理、即ち、ユーザ・アプリケーションの実行を瞬間的に停止する、図50に示されたステップのシーケンスに移る。次いで、オン・ライン・プロセッサ20の残りの状態（例えば、構成レジスタ、キャッシュ、等）及びそのキャッシュは、読取られかつメモリ28のバッファ（メモリ位置のシリーズ）に書込まれる（ステップ1102）。次いで、その状態は、両方のCPU s 12A, 12Bのプロセッサ装置20をリセット命令に指向する“リセット・ベクトル”と共に、オフ・ラインCPU 12Bにコピーされる。次に、ステップ1106は、SLEEP記号によってルータ14A, 14Bを静止し、ルータのFIFOsがクリアであり、プロセッサ・インタ

ーフェイス24のFIFOsがクリアであることを確実にすべく自己アドレス指定されたメッセージ・パケットがそれに続き、そして更なる入力I/Oメッセージ・パケットが今後来ない。ステップ1108では、オン・ラインCPU 12Aは、SRST記号を両方のCPU s 12A, 12Bへエコーバックするルータ14A, 14BにSRST指令記号を送信する。エコーバックしているルータは、上述したスレーブ・デブプレックス・モードでまだ動作しているので、オフ・ラインCPU 12BにエコーバックされたSRSTは、オン・ラインCPU 12Aへエコーバックされたその後まだ8クロックである。エコーバックされたSRST記号は、各CPUのプロセッサ装置20を、リセット・ベクトルを含むメモリ28の位置へジャンプさせかつプロセッサ装置20、キャッシュ22、レジスタ、等へ両方のCPU s 12A, 12Bの記憶された状態を復元するサブルーチンを起動させるべく、受信されかつ両方のCPU s 12A, 12Bにより作用される。次に、CPU s 12A, 12Bは、同じ命令ストリームを実行し始める。

【0244】それゆえに、ステップ1112で、CPU s 12A, 12Bは、まだシャドウ・モード動作である、即ち、両方は、同じ命令ストリームを実行しているが、CPU 12Bは、8クロック・サイクルCPU 12Aに遅れてそのように行っており、ルータ14は、CPU 12Bからの送信を無視すべくまだ構成される。CPU 12Aは、ユーザ・アプリケーションの実行を再開すべくオン・ライン状態へ戻る。再統一手順は、図51に示したような、“怠惰な再統一（lazy reintegration）”と呼ばれる、再統一の最終ステージにいまここに入る。汚れた位置をマークするためのECCビットのイネープリングは、プロセッサが同じメモリに対して同じ事を行っているので、いまここでディスエーブルされなければならない。再統一手順のこのステップの間中、オン・ラインCPU 12Aが、命令を実行しているときに（オフ・ラインCPU 12Bも実行している - 8クロックの遅延にもかかわらず）メモリを読取っているときに汚れたとマークされたメモリ28の位置に出会うときには、それは、“バス誤り（bus error）”を起動する（ステップ1120）。このバス誤りの表示は、同じ命令に対してオフ・ラインCPU 12Bにバス誤りを強要すべく“ソフトフラグ（soft-flag）”論理回路素子900（図44）の選択論理回路920を用いて、CPU 12Bへ送信される（ステップ1122）。図44をちょっと参照すると、REINTが、MUX 914を介して、バス誤りがCPU 12Aによって出会ったことをCPU 12Bに知らせるためにCPU 12BへのBUS ERROR信号を選択することをアサートしているということが理解できる。

【0245】その間に、CPU 12Aのバス誤りは、
（1）誤りの原因及び（2）もし可能ならば誤りの処理

方法を決定すべくプロセッサ装置20を誤り処理ルーチンに強要させる。この場合、誤りが、汚れたとマークされたメモリ位置を読取る試みによりもたらされたということが決定される。従って、プロセッサ装置20は、CPU12Bへメモリ位置の内容をコピーすべくAtomic Write機構を(BTE88を介して一図9)起動する。次に、CPU12Aは、バス誤りをもたらし命令を再実行し、かつ進む。CPU12Aに8クロック・ステップ遅れて動作しているCPU12Bも、CPU12Aにバス誤りをもたらし命令の実行に先駆けて、バス902を介してCPU12Aからのその誤りの伝達によって強要されたバス誤りを有する。しかしながら、CPU12Bがその命令を実行するときまでには、バス誤りの表示は、CPU12Bへ伝達されかつ8クロック後にCPU12Bの同じ命令と相関される。この相関関係は、オン・ラインCPU12Aからオフ・ラインCPU12Bへバス誤り信号をバスすることにおける遅延を、CPUsへのルータ送信によって導入された8クロック遅延(即ち、シャドウ・モードの8クロック遅延)とマッチングすることによって達成される。しかしながら、CPU12Bは、CPU12Aが起動した同じバス誤り処理ルーチンを通して行くことを強要される。ロックステップ同期動作に留まるために、オフ・ラインCPU12Bは、バス誤りルーチン及び“汚れた”メモリ位置からルータへデータを送信することを含んでいる、オン・ラインCPU12Aとまったく同じ動作のシーケンスを実行する。ルータは、CPU12Bの送信を無視するが、CPU12Bは同じ動作を行うためにCPU12Aによって費やされたのと同じ時間の量を費やさなければならないということに留意する。

【0246】その間に、オン・ラインCPU12Aは、CPU12Aのメモリ全体を通して一つの最後のバスを行うためにある時間を割り当てて、ユーザのアプリケーション・プログラムの実行を続けている間に、まだ汚れたとマークされているそれらのメモリ位置にわたってコピーする。この再統一の最後のステージの間中、メモリ全体は、全てのメモリ位置を検査すべく読取られる。検査されかつ汚れたとマークされるべきであることが見出された全ての位置は、オフ・ラインCPU、CPU12Bへコピーされる。最後に、CPU12A、12Bの状態は、二つのCPUsが真性の、遅延されない、ロックステップ動作に置かれることができるように一致する。それゆえに、一度、本当に、全てのメモリが検査され、かつ必要ならば、コピーされるということがステップ1124で決定されたならば、ステップ1128では、MP18は、制御論理回路509に含まれた構成レジスタへ書き込むことによってデュプレックスの次のモード状態にルータ14Aをセットする。次に、CPU12Aは、以前のようにSLEEP、自己アドレス指定されたメッセージ・パケット・シーケンスを発行する。CP

U12Aがルータが静止状態であるということを実際にしたときに、CPU12Aは、SRST記号を両方のルータ14A、14Bへ送る(同時に)。ルータ14A、14Bによるその記号の受信は、それがSRST記号を二つのCPUs12A、12Bへエコーバックしたときに、それらが両方とも同時にエコーバックされようとしてそれらをデュプレックス・モードに移動する。SRST記号がCPUs12A、12Bによっていま受信されるときに、それらは、CPUsの両方のプロセッサ装置20を、同じ仮想時間で同じ状態を有する同じ位置からスタートすべくリセットさせる。CPUs12A、12Bは、いまロックステップ動作である。

【0247】追加機能

一 低減コスト・デュプレックス・システム

図1～図3をちょっとの間考慮すると、指摘されたように、CPUs12A、12Bは、別々に、またはデュプレックスされたペアのいずれでも用いる。前者の場合には、各個別に動作するCPUの設計に用いた冗長は、フェイルファースト・アーキテクチャを供給する。CPUsは、フォルトトレランスに対してソフトウェア・アプローチを実施すべく一つのCPUが“主(1次)”CPUに、他のCPUが“2次”CPUに指定されるように(デュプレックスされるのではなく)ペアにされる。それゆえに、2次CPUは、主CPU上で走っているユーザ・アプリケーションを利用可能であり、かつ主CPUは、例えば、データ・ベースまたは更新時におけるその地点までの主CPUの処理の表示である監査ファイルを周期的に更新する。主CPUが故障したならば、2次CPUがバックアップ・アプリケーションを起動しかつデータ・ベースまたは監査ファイルが最後に更新された時点から故障したCPUに取って代わる。これが、ソフトウェア・フォルトトレランス・アプローチである。

【0248】ソフトウェア・フォルトトレランス・アプローチは、オペレーティング・システムによって一般に実施される。頑強でなく、従って、この能力(容量)を有していないようなそれらのオペレーティング・システムに対しては、上述したデュプレックスされた動作のモード(図1～図3を参照)が与えられ、二つのCPUs12を用いて同じ命令ストリームの同じ命令を実行すべく動作する。図52に示したのは、低減コスト・デュプレックスされたCPUsのペアであり、その一つが他の冗長を有していない。しかしながら、図1～図3を参照すると、CPU12Aは、両方がデュプレックスされかつロックステップ・モードで動作しているときに

CPU12Aの個々のプロセッサ装置20a、20bがCPUに対してフェイルファースト、フォルトトレランスを供給するのと同じ方法で、CPU12Bに対する誤り検査冗長で動作することができることに注目する。それゆえに、図52に示したように、デュプレ

ックスされた動作に対して、低減コスト・システムが適用可能である。図52に示したように、処理システム10'は、上述したように構成されたCPU12Aとルータ14A、14Bを含む。ここでCPU12B'として示された、CPU12AがペアになるCPUは、しかしながら、単一マイクロプロセッサ・ベースCPUシステムとして構成される。また、ルータ14A、14BとCPUsとの間の接続は、同じである。

【0249】それゆえに、CPU12B'は、単一プロセッサ装置20'及び、キャッシュ22'、インターフェイス装置(IU)24'、メモリ・コントローラ26'、及びメモリ28'を含んでいる、関連支持コンポーネントだけを備えている。それゆえに、CPU12Aは、キャッシュ・プロセッサ装置、インターフェイス装置、及びメモリ制御冗長を伴って、図4に示した方法で構成されると同時に、それらのおおよそ半分がCPU12B'を実施するために必要である。動作において、CPU12A、12B'は、デュプレックス・モードで動作され、それぞれが、同一の命令ストリームの、同じ命令を実質的に同時に実行する。CPU12Aは、プロセッサ装置20及びCPUを構成する他の素子の複製を通してフェイル・ファースト動作を供給すべく設計される。更に、発散に対してルータ14A、14Bによってなされたデュプレックス動作及び検査を通して、CPU12Aは、また、そのコンパニオンCPU、CPU12B'に対するチェック・アップを供給する。ペアによって形成された論理CPUは、発散がルータ14A、14Bの一つによって検出されかつ発散の検出が故障しているCPUを停止すべく上述したように作用されるならば、残りのCPUsがアプリケーションを続行することができるようなフェイル機能動作を供給する。

【0250】残りのCPUが12Aであるならば、CPU12Aを構成する複製されたコンポーネントによる少量のデータ・インテグリティがまだ存在する。生き残っているCPUがCPU12B'であるならば、通常ファッションで実施された誤り検査(即ち、種々のインターフェイスでのパリティ検査)を除き、データ・インテグリティが欠けているということが認識される。図52は、二つのCPU12A、12B'から出力されたデータの比較を実行すべく一対のルータ14A、14Bが含まれて、処理システム10'が示されている。しかしながら、発散検査だけが実行されるべきであるならば一つのルータ14だけが(例えば、ルータ14A)用いられるのが必要である。事実、CPU12A、12B'から出力されたデータを受信すべく接続された二つの入力、出力されたデータのある程度非同期的な受取りを受信すべく上述したようなクロック同期FIFOsを有し、同期ファッションでFIFOsから出力したその受信したデータをプリングする、ということを定めて、ルータの使用は、発散に対して必要な検査を実行すべく簡

単なコンパレータ回路以外の何者でもないもので置き換えられるということは、当業者には明らかである。

【0251】— スタンバイ・スペアリング(Standby Sparring)

図1~図3をちょっと参照すると、これらの図に示された処理システムのアーキテクチャの重要な特徴は、各CPU12がそれに利用可能な全てのI/O Packet Interface (パケット・インターフェイス) 16のサービス、及びシステムにおける他のCPU12の援助(支援)なしで、装着されたI/O装置を有することである。多くの従来の並列処理システムは、特定のプロセッサまたはCPUの援助によってのみI/O装置へのアクセスまたはそのサービスを供給する。そのような場合、I/O装置のサービスに対して責任があるプロセッサが故障したならば、I/O装置は、システムの残りに対して利用不可能になる。他の従来のシステムは、プロセッサの一つが故障したならば、対応I/Oへのアクセスが残りのI/Oを通してまだ利用可能であるようにプロセッサのペアを通してI/Oへのアクセスを供給する。もちろん、両方が故障したならば、再度I/Oは、失われる。また、並列またはマルチ処理システムの他のプロセッサを供給するためにプロセッサの資源を必要とすることは、システムに性能インパクトを課す。

【0252】全ての周辺装置へのアクセスをマルチ処理システムの全てのCPUに許容する能力(機能)は、ここでなされたように、上記した米国特許第4,228,496号公報に教示された“主/バックアップ”処理を拡張すべく動作する。そこでは、多重CPUシステムは、バックアップ処理が別のCPUsのバックグラウンドに存在すると同時に、一つのCPU上で走らせる主処理を有する。周期的に、主処理は、処理の動作に関するデータがバックアップ処理にアクセス可能な位置に記憶されるような“チェックポインティング(check-pointing)”動作を実行する。主処理を走らせているCPUが故障したならば、その故障は、バックアップが存在するものを含んでいる、残りのCPUsによって検出される。CPU故障のその検出は、バックアップ処理を起動させ、かつチェックポイント・データをアクセスするために、バックアップに最後のチェックポイント動作の地点から前の(former)主処理の動作を再開させる。バックアップ処理は、いま主処理になり、残っているCPUsのプールから、新しい主処理のバックアップ処理を有すべく一つが選ばれる。従って、システムは、最初の故障(即ち、故障したCPU)が修復される前でさえも別の故障を許容できるような状態に素早く復元される。

【0253】それゆえに、処理システム10の種々の素子を相互接続する方法及び装置は、全てのCPUに、そのシステムの全てのI/O素子、並びにシステムの全てのCPUへのアクセスを供給するということが理解できる。各CPUは、別のプロセッサのサービスを用いる必

要なしにあらゆる I/O をアクセスできる。それにより、システム性能は、I/O をアクセスすることに含まれる特定のプロセッサを必要とするシステムに比べて向上されかつ改良される。更に、CPU 12 が故障し、またはラインから取り除かれても、そのアクションは、他の CPU のシステムの I/O へのアクセスになんの影響を及ぼさない。

【0254】— トランザクション・シーケンス・プロトコル及びバリア・トランザクション：上述したように、パケットのヘッダ・フィールドは、4 ビット・トランザクション・シーケンス番号(Transaction Sequence Number) (TSN) フィールドを含む；図 5 (a) 及び図 5 (b) を参照。CPU 12 A またはある I/O 装置のような、二つ以上の顕著な(outstanding) 要求を管理するように構成された処理システム 10 の素子(図 1～図 3)は、TSN フィールドにおいて各顕著な要求に対して固有のシーケンス番号を供給する。宛先素子が特定の要求に対する応答パケットを生成するときに、応答パケットの TSN フィールドは、応答を促した要求パケットにおけるのと同じ TSN 値を含む。次に、応答を受信するシステム素子は、どの要求に応答が対応するかを決定すべく応答における TSN にマッチすることができ。TSN は、応答がもはや存在しない要求に答えるかどうかをシステム素子に決定させる。例えば、このように、あるシステムは、要求に対する応答に所定の期間内に受信されることを要求する。予期したようには応答が受信されなかったならば、要求を起動したシステム素子は、第 2 の(再度の)要求を単に発行する。早めの要求に対する応答がその後に受信されたならば、システム素子は、応答が答える要求(早めの、無効の、要求、または遅く有効な要求)を TSN から決定することができる。前者ならば、応答は、廃棄される。

【0255】TSN's は、通称“陳腐なパケット(stale packet)”問題を処理することも支援する。誤りが発生するときに、通過中のメッセージ・パケットは、ネットワークのどこかで動かなくなりうる。これら陳腐なパケットを除去する方法が存在しないならば、それらは、後で現れることができかつシステムが最初の問題から回復した後に動作をもしかすると破壊する。受信した応答メッセージ・パケットの TSN は、受信機に、応答によって運ばれた TSN を応答を促したメッセージ・パケットの TSN と比較することによって応答が現行であるか否かを決定させる。小さな TSN を用いることは、現在顕著である要求にマッチしうる TSN を伴って陳腐な(stale) 応答が後で現れるという可能性を生ずる。しかし、大きな TSN フィールドは、伝達されたメッセージ・パケットのそれぞれがより大きいこと、またはそれによってデータ・フィールドが低減されることのいずれかを要求する傾向がある。本発明は、“バリア・トランザクション(Barrier Transaction).”と呼ばれる機構を通

してこの問題を解決する。TSN's は、用いられるべく継続するが、Barrier Transaction 機構は、TSN の必要なサイズをほんの 4 ビットのフィールドまで低減する。

【0256】簡単に言うと、Barrier Transaction は、送信ノードと受信ノードとの間の通信経路のインテグリティを検査するために用いられる。それが I/O インターフェイス 16 によって発行されうけれども、Barrier Transaction は、CPU によって主に起動される。それは、I/O 装置 17 または CPU 12 向けの先に発行されたメッセージ・パケットに対する予期した応答が所定の割当て期間内に受信されないときに主に用いられる。CPU 12 は、通常のヘッダ、アドレス、データ、及び CRC フィールドを含んでいる、HADC パケット(図 5 (a))の形の Barrier Transaction メッセージ・パケットを生成しかつ送ることによって経路を検査できる。Barrier Transaction メッセージ・パケットによって運ばれたデータは、トランザクションを独自に識別し、かつそのデータのコピーは、CPU による後の比較のために CPU によって保管される。Barrier Transaction メッセージ・パケットを受信しているシステム素子(例えば、それが別の CPU でもありうるが、I/O インターフェイス 16 の一つ)は、Barrier Transaction 応答を生成しかつ送ることを要求される。しかしながら、そのようにする前に、Barrier Transaction 応答は、それが Barrier Transaction に応答できる前に Barrier Transaction メッセージ・パケットの受信に先駆けて受信した(要求を発行したシステム素子からの)全ての要求を終了または廃棄することを要求される。Barrier Transaction 応答は、Barrier Transaction 要求で運ばれた同じデータを含んでいる、HDC 形(図 6)のものである。Barrier Transaction 応答がトランザクションを起動した CPU によって受信されたときに、応答におけるデータは、応答が対応する Barrier Transaction (多数の顕著な Barrier Transaction が存在しうる)を特に決定すべく、(CPU によって早めに保管された)早めに送られた Barrier Transaction メッセージ・パケットに存在したデータと比較される。

【0257】システム素子と他のシステム素子(例えば、CPU 12 A と I/O 17_n；図 1～図 3)の間にただ一つの有効経路が存在するので、かつメッセージ・パケットがその宛先へ向かう途中で他のメッセージ・パケットを渡すことができないので、メッセージ・パケット受信のシーケンスは、それらが送られたようなシーケンスにマッチする。それゆえに、Barrier Transaction は、Barrier Transaction を発行しているシステム素子と Barrier Transaction を受信しかつそれに応答しているシステム素子との間の経路をクリアすべく動作する。Barrier Transaction 応答の受信は、全ての要求は Barrier Transaction が答えられる前に送るか、またはやつ

て来ないかのいずれかであることをBarrier Transactionを発行したシステム素子に知らせる。それゆえに、無回答要求が再発行され、応答が最終的に受信されたならばそれは、再発行された要求の結果であり、早めの（及び先に無回答であった）要求に対する遅延応答でないということを知る。Barrier Transaction 機構は、ほんの2～3のTSN番号の使用を許すということが理解できる。（ここでは、ある程度大きなフィールドを必要とする従来システムに対して、ほんの4ビットのTSNフィールドが用いられる。）

バリア・トランザクションの動作は、カスケード・ルータ14A及び14X、を含むX経路によりI/Oパケット・インターフェイス16Aに結合されたCPU12Aと、TNetリンクL（即ち、リンクL_X、L（1）、及びL）を示す図53に示されている。上述したように、各ルータは、エラスティックFIFOs 506を含むポート入力502を有する。この説明に対して、エラスティックFIFOだけが必要であり、従って示されている。

【0258】ルータ14A及び14Xとリンク・セクションL（1）'との間のリンクL（1）のアウトバウンド・セクションは、図53において破線で示したように、使用不可能になると想定する。これは、多数の理由で発生しうる：故障しているコネクタ、ずれたケーブル、等。ルータ14Aからルータ14Xへのアウトバウンド・メッセージ・トラフィックは、中断する（やめる）。I/Oパケット・インターフェイス16Aに向かう途中であるが、故障しているリンク・セクションL（1）'のまだ上流であるCPU12Aによって起動されたメッセージ・パケット・トランザクションは、応答されず、従って、通信経路における故障を示すことが時間切れになる。割込みが内部的に生成され、かつプロセッサ20（20a, 20b - 図4）は、バリア要求（BR）ルーチンの実行を起動する。そのバリア要求ルーチン（BR）は、応答の欠如に対して時間切れになる各発行されたトランザクション（メッセージ・パケット）に対する各AVTエントリ（図18）の許可フィールド（図19）におけるPEXビットをクリアすることによって経路をまずディスエーブルする。これは、顕著なトランザクションによって促された応答メッセージ・パケットが後で現れたならば、それは、AVTエントリがその応答に対してアクセスされかつ検査されたときに阻止されることを確実にする；即ち、リンクにおける故障の理由で失速されたのではなく、それらが最終的に宛先に到達する前に一時的に無くなる、メッセージ・パケット。

【0259】ある後の時間に、リンクL（1）は、修正され、かつルータ14AのエラスティックFIFOs 506'における1152で示されたような、いまでは陳腐な(stale) メッセージ・パケットを解放する。リンク

L（1）の再制定は、CPU12AがそれからI/Oパケット・インターフェイス16AへのX経路がいま動作に戻るという可能性をいま認識しているようにMP18によってCPU12Aに知らされる。しかしながら、CPUは、（I/Oパケット・インターフェイス16Aに対応している適切なAVTエントリにおけるPEXビットをリセットすることによって）その経路をまだイネーブルできない。理由は、I/Oパケット・インターフェイス16Aに、全く異なるメッセージ・パケットとしてそれを誤解させて、それによって応答させるべくその最初の宛先（I/Oパケット・インターフェイス）へフローすべく継続している、エラスティックFIFOs 506'において1152で示されたような、陳腐なトランザクション・メッセージ・パケットの可能性である。この問題を防ぐために、かつCPU12AによってX経路が通常のトラフィックに再び用いられる前に、CPU12Aで実行しているBRサブルーチンは、“Barrier Request（バリア要求）”メッセージ・パケットをI/Oパケット・インターフェイス装置16Aに送ることによってBarrier Transactionを起動すべくBTE論理回路88（図9及び図23も参照）を用いる。Barrier Requestメッセージ・パケットは、メッセージ・パケットのヘッダ（図5（a）及び図5（b）参照）に含まれるソース・フィールドのサブフィールドによってそのように識別される。上記したように、Barrier Requestメッセージ・パケットのデータ・フィールドは、その特定のトランザクションに対して固有なデータ値を含む。

【0260】Barrier Requestメッセージ・パケット（即ち、1150）が、I/Oパケット・インターフェイス16AのXインターフェイス装置16aによって受信されたときに、それは、そのデータ・セクションが受信したBarrier Requestメッセージ・パケット1150に含まれた同じ同一で、固有なデータ値を含む、応答メッセージ・パケットをフォーミュレートする。次に、I/Oパケット・インターフェイス装置16Aは、ルータ14X、14Aを介して、CPU12Aに応答を戻し送信する。バリア要求メッセージ・パケットへの応答がCPU12Aによって受信されるときに、それは、AVT論理回路90'を通して処理される（図9及び図16も参照）。バリア応答は、他の型のトランザクション以外のバリア応答を終了させるべくエントリの対応許可フィールドにセットされた“B”フィールドを有するAVTエントリを用いる。（Barrier Transactionが送られたときに、AVTエントリは、応答を確認するのに用いるためにCPUによって生成された。）

上述したように、各バリア・トランザクションは、それに応じて送信者に戻されるデータ値を含む。この固有な値は、CPU（即ち、BRルーチン）に、送られたデータ値を応答で受信したものと比較させて、応答が異なるバリア・トランザクションの一部でなかったことを確実

にする。一度、バリア応答がそれをCPU 12Aに戻せると、陳腐なパケットがこの経路に沿ってFIFOバッファに残るという可能性がもはや存在しない。また、CPU 12Aは、先にディスエーブルされた経路が通常のトラフィックに対して再び用いることができるということを制定した。従って、CPU 12Aは、その経路を用いる全てのAVTエントリにおけるPEX許可フィールドをセットすることによって経路を再イネーブルする。

【0261】本発明の全て及び完全な開示がなされたが、種々の代替及び変更が特許請求の範囲の真の範囲から逸脱することなく本発明の種々の態様に対してなされうということが当業者に明らかになるであろう。例えば、ある一定の誤りを検出できる8ビット/9ビット・コードの形の指令/データ記号の送信において発生する誤りの検出を供給するためのスキームが開示された。概念は、9ビット/10ビット・コード、または、多重バイト・ワイドのような他の同様なコードへ更に導くことができるということが当業者に明らかであるべきである。更に、ルータ14は、あらゆる数のポートを有すべく構成されうる；指令/データ・パケット・フォーマットは、(ヘッダ、及び他のフィールドにより多くのまたはさらに少ないビットを有して)異なりうる；ルーティング・トポロジーは、ルータ14を用いて、リング、木、ハイパーキューブ、等として形成することができる。

【図面の簡単な説明】

【図1】本発明の教示に従って構成された処理システムを示す図である。

【図2】図1の処理システムのクラスタまたは配置構成を採り入れている図1の処理システムの代替構成の一つを示す図である。

【図3】図1の処理システムのクラスタまたは配置構成を採り入れている図1の処理システムの代替構成の別の一つを示す図である。

【図4】図1～図3の各サブプロセッサ・システムの一部を形成する中央処理装置(CPU)を示す略ブロック図である。

【図5】図5は(a)～(d)からなり、図4に示すエリア・ネットワークI/Oシステムを介して入力/出力データのような情報を運ぶために用いられる種々のメッセージ・パケットの構成を示す図である。

【図6】図4に示すエリア・ネットワークI/Oシステムを介して入力/出力データのような情報を運ぶために用いられる種々のメッセージ・パケットの一構成を示す図である。

【図7】図4に示すエリア・ネットワークI/Oシステムを介して入力/出力データのような情報を運ぶために用いられる種々のメッセージ・パケットの他の一構成を示す図である。

【図8】図4に示すエリア・ネットワークI/Oシステ

ムを介して入力/出力データのような情報を運ぶために用いられる種々のメッセージ・パケットの別の一構成を示す図である。

【図9】プロセッサ及びメモリをI/Oエリア・ネットワーク・システムとインターフェイスするための図4のCPUの一部を形成するインターフェイス装置を示す図である。

【図10】図9のインターフェイス装置のパケット受信機の部分を示している、ブロック図である。

【図11】図10に示すパケット受信機のパケット受信機セクションに用いられるクロック同期FIFO(CS FIFO)を示す図である。

【図12】図11に示すクロック同期FIFO構造の構成のブロック図である。

【図13】CPUの二つのインターフェイス装置からの誤り検査外向き送信に対するクロス接続を示す図である。

【図14】エンコードされた(8Bから9B)データ/指令記号を示す図である。

【図15】データがデータ誤り検査のためにメモリコントローラに転送される誤りのクロスチェックのために図9のインターフェイス装置によって用いられる方法及び構造を示す図である。

【図16】処理システムの他の(CPUの外部の)コンポーネントに図4のCPUのメモリへの読取り及び/又は書込みアクセスをスクリーンしかつ付与するために用いられるアクセス妥当性検査及び変換(AVT)表の実施を表わすブロック図である。

【図17】AVT表エントリをアクセスするために用いられるアドレスの形成を示すブロック図である。

【図18】通常及び割込み要求に対するAVT表エントリを示す図である。

【図19】通常及び割込み要求に対するAVT表エントリを示す別の図である。

【図20】通常及び割込み要求に対するAVT表エントリを示す別の図である。

【図21】メモリのキューに対する及び図4のCPUのプロセッサ装置に対する割込み要求をポスティングする論理回路を示す図である。

【図22】キュー・エントリに対するメモリ・アドレスを形成するために用いられる処理を示す図である。

【図23】プロセッサ装置によって図4のCPUのメモリに形成され、かつ図1～図3に示すエリアI/Oネットワークを介して送られるべきデータを含んでいる、データ出力構成を示し、かつ図11、12のパケット送信機セクションを通るエリアI/Oネットワークへの送信に対するデータ出力構成をアクセスすべく動作する図9のインターフェイス装置のブロック転送エンジン(BTE)装置も示すブロック図である。

【図24】8チェック・ビットと共に連続偶数アドレス

で二つの同時にアクセスされた32ビット・ワードを含んでいる、データの72ビットをメモリからアクセスするために図4のCPUのメモリとそのインターフェイス装置との間で一对のメモリ・コントローラによって部分形成される72ビットのデータ・パスの構造を示す図である。

【図25】オンライン・アクセス・ポート（OLAP）を通るそれへの順次アクセスを示している、図4に示す二つのメモリ・コントローラの一つの略ブロック図である。

【図26】図4の一对のメモリ・コントローラの状態マシン及び誤り検査のために一つを他のものに対して検査するために用いる技術を簡略形で示す図である。

【図27】図1～図3に示す処理システムのエリア入力／出力ネットワークに用いるルータ装置を示す略ブロック図である。

【図28】図27に示すルータ装置の二つのポート入力の比較を示す図である。

【図29】図27に示すルータ装置の6つの入力ポートの一つの構造を示すブロック図である。

【図30】図27のルータ装置の入力ポートで受信する指令／データ記号の妥当性を検査するために用いる同期論理回路のブロック図である。

【図31】図29に示す入力ポートのターゲット・ポート選択論理回路を示すブロック図である。

【図32】図31のターゲット・ポート選択論理回路によって行われるルーティング決定を示している決定チャートである。

【図33】図31のターゲット・ポート選択論理回路の一部を形成するアルゴリズム・ルーティング論理回路のブロック図である。

【図34】図27に示すルータ装置の6つの出力ポートの一つを示すブロック図である。

【図35】図11の一对のFIFOsを用いて（各CPUに対して一つ）、処理システムがロックステップ（デュプレックス）モードで動作しているときに同期したファッションで図4のデュプレックス・ペアCPUsに同じ情報を送信するために用いられる方法を示す図である。

【図36】そのサブ処理システムの種々の素子を動作するために用いられる複数のクロック信号を開発するための図1～図3のサブ処理システムのそれぞれのクロック生成システムを示している略ブロック図である。

【図37】互いに一对のサブ処理システムの種々のクロック信号を同期するために対になったサブ処理システムのクロック生成システムを相互接続するために用いられるトポロジーを示す図である。

【図38】FIFOのキューに記号を押し付けかつそれらを引き離すために用いられる二つのクロックがかなり異なるときの状況において図13または図29及び図3

0のクロック同期FIFOを制御するために用いられるFIFO一定速度クロック制御論理回路を示す図である。

【図39】FIFOのキューに記号を押し付けかつそれらを引き離すために用いられる二つのクロックがかなり異なるときの状況において図13または図29及び図30のクロック同期FIFOを制御するために用いられるFIFO一定速度クロック制御論理回路を示す別の図である。

【図40】図38及び図39の一定速度制御論理回路の動作を示すタイミング図である。

【図41】素子を構成するために図1の（または図2または図3に示す）システムの種々の素子に対する保守プロセッサ（MP）へのアクセスを供給するために用いられるオンライン・アクセス・ポート（OLAP）の構造を示す図である。

【図42】キャッシュ・ブロック境界を示している、システム・メモリの部分を示す図である。

【図43】デュプレックス・モードで動作する対になったサブ処理システムのCPUs間の非同期可変を処理するために用いられるソフト・フラグ論理回路を示す図である。

【図44】デュプレックス・モードで動作する対になったサブ処理システムのCPUs間の非同期可変を処理するために用いられるソフト・フラグ論理回路を示す他の図である。

【図45】互いに情報を受け取る図1の処理システムのCPUs及びルータのクロック同期FIFOsをリセットしかつ同期するために用いられるフロー図である。

【図46】互いに情報を受け取る図1の処理システムのCPUs及びルータのクロック同期FIFOsをリセットしかつ同期するために用いられるSYNC CLKの部分を示す図である。

【図47】デュプレックス・モードで動作している二つのCPUs間の相違（ダイバージェンス）を検出しかつ処理するために用いられる手順を概略的に示している、フロー図である。

【図48】他のCPUsが処理システムの動作を計れる程度に停止することなく図1に示す処理システムのCPUsの一つをロックステップ、デュプレックス・モード動作にするために用いられる手順を一般に示す図である。

【図49】他のCPUsが処理システムの動作を計れる程度に停止することなく図1に示す処理システムのCPUsの一つをロックステップ、デュプレックス・モード動作にするために用いられる手順を一般に示す他の図である。

【図50】他のCPUsが処理システムの動作を計れる程度に停止することなく図1に示す処理システムのCPUsの一つをロックステップ、デュプレックス・モー

ド動作にするために用いられる手順を一般に示す他の図である。

【図51】他のCPUが処理システムの動作を計れる程度に停止することなく図1に示す処理システムのCPUの一つをロックステップ、デュプレックス・モード動作にするために用いられる手順を一般に示す他の図である。

【図52】本発明の教示を組み込んで低減コスト・アーキテクチャを示す図である。

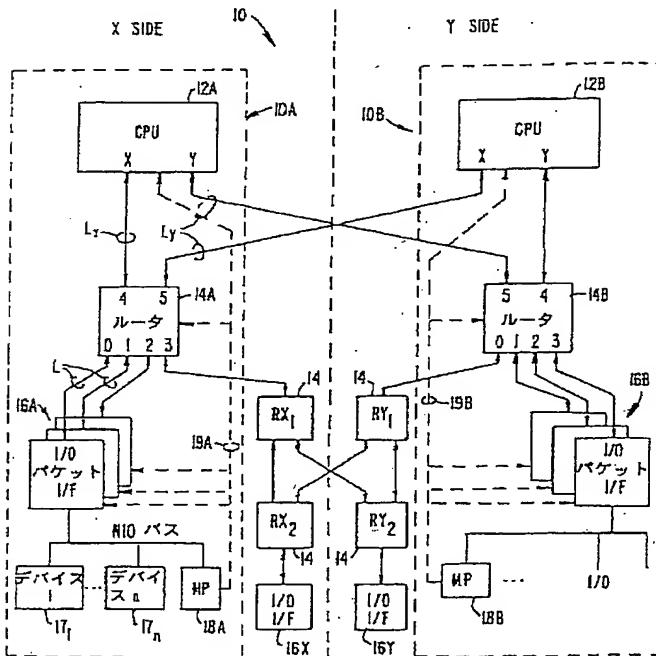
【図53】図1（または図2、図3）のCPUと入力／出力装置との間の通信経路を検査しかつ検証するバリア

・トランザクションの動作を示す図である。

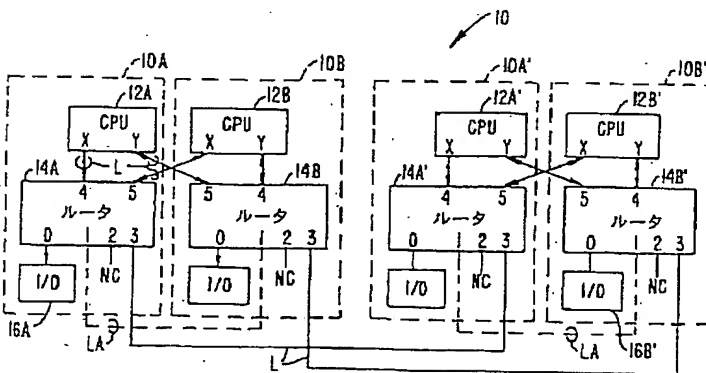
【符号の説明】

- 10 データ処理システム
- 10A, 10B サブプロセッサ・システム
- 12A, 12B 中央処理装置 (CPU)
- 14, 14A, 14B ルータ
- 16A, 16B, 16X, 16Y I/Oパケット・インターフェイス
- 17₁ ~ 17_n I/O装置
- 18A, 18B 保守プロセッサ (MP)

【図1】



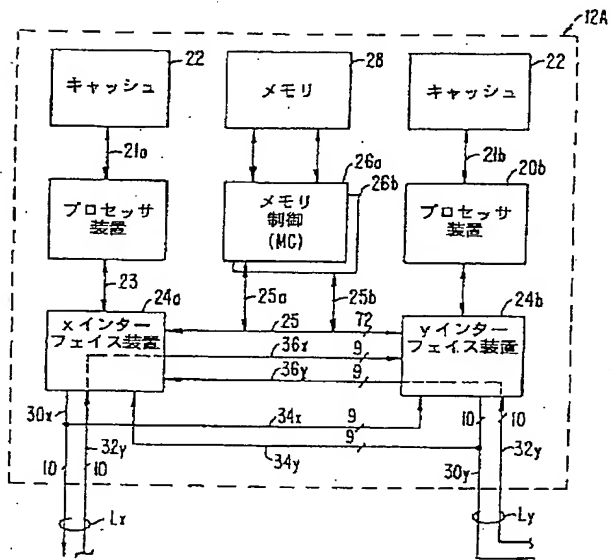
【図2】



【図14】

CDC	CDB	CDA	CDS	CD4	CD3	CD2	CD1	CDO
-----	-----	-----	-----	-----	-----	-----	-----	-----

【図4】



【図6】

【図7】

HAC パケット・フィールド

ヘッダ	アドレス	CRC
バイト 8	4	4 = 16

HDC パケット・フィールド

ヘッダ	データ	CRC
バイト 8	N	4 = 12+N

【図8】

HC パケット・フィールド

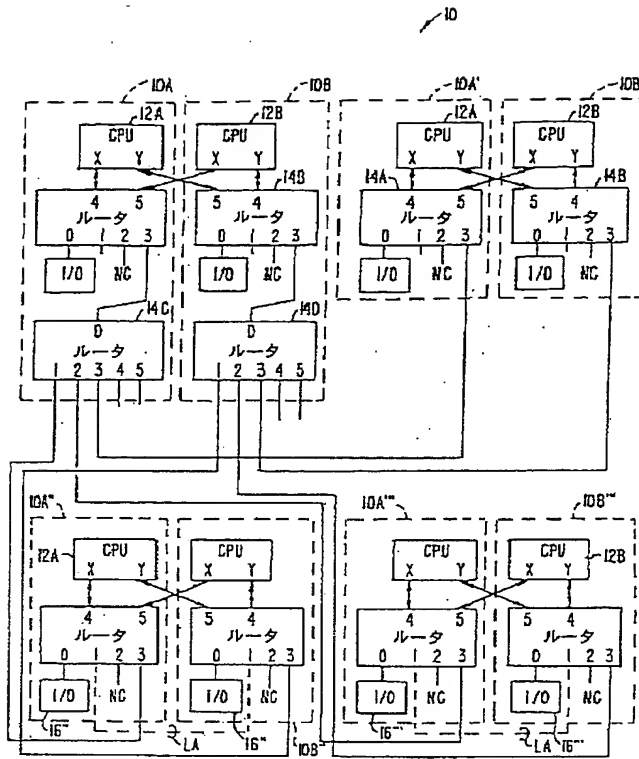
ヘッダ	CRC
バイト 8	4 = 12

【図19】

AVT 許可フィールド

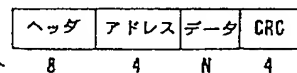
E	PEX	PEY	I	CI:OJ	W	R	B	RSVD
---	-----	-----	---	-------	---	---	---	------

【図3】

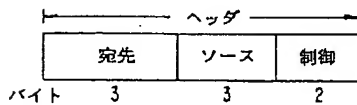


【図5】

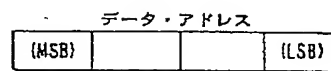
HADC パケット・フィールド



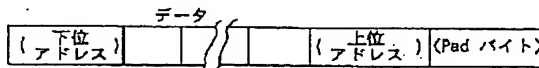
(a)



(b)



(c)



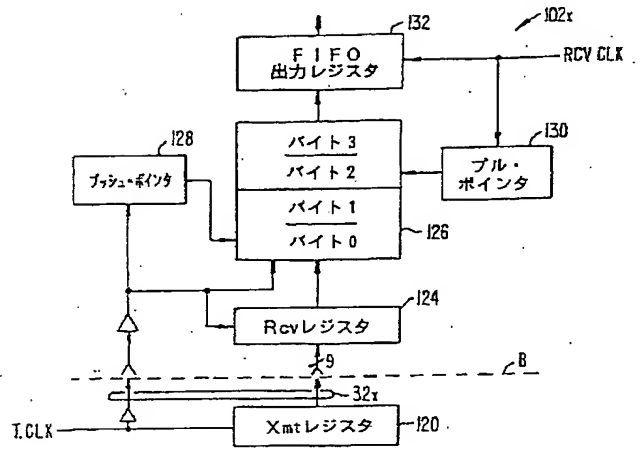
(d)

【図18】

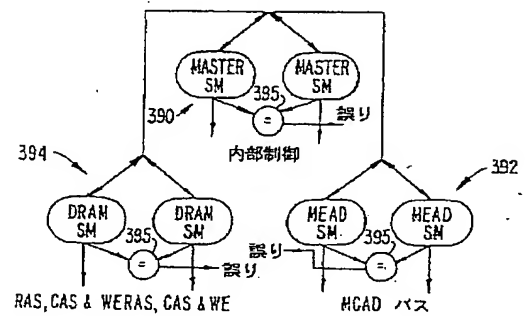
AVT エントリ (標準)

20ビット	52ビット	12ビット	12ビット	20ビット	12ビット
(リザーブ ド)	物理頁番号	下部バウンド	許可	ソースID	上部バウンド

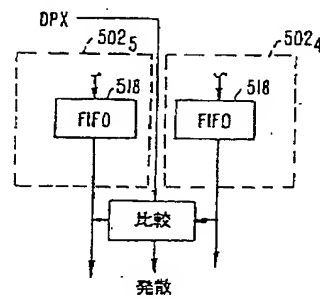
【図11】



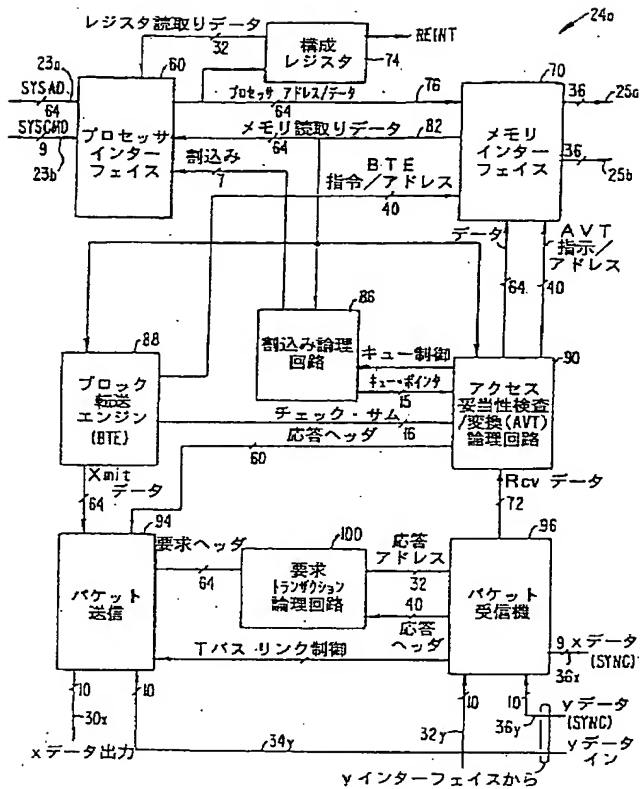
【図26】



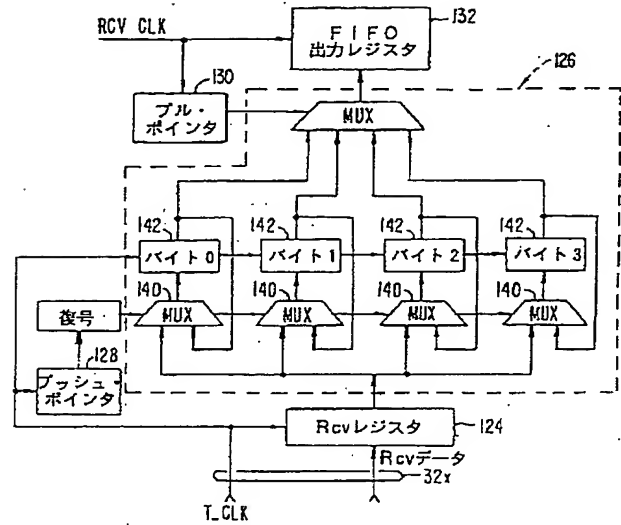
【図28】



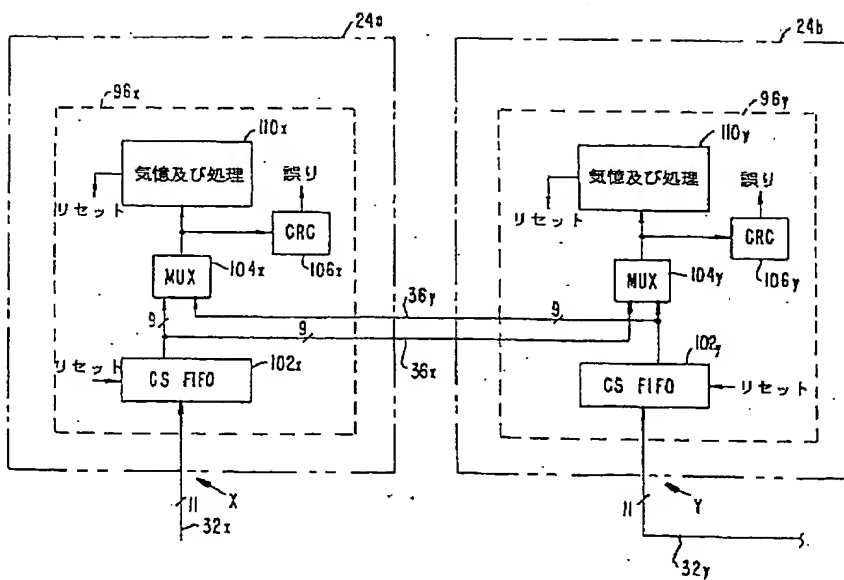
【図9】



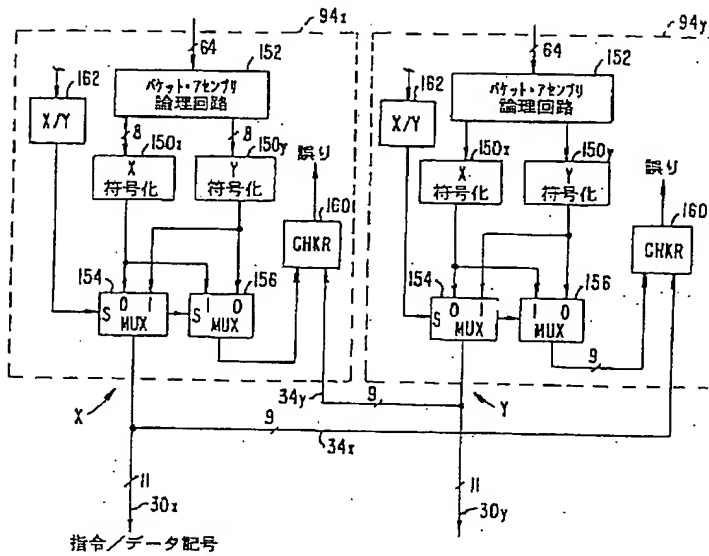
【図12】



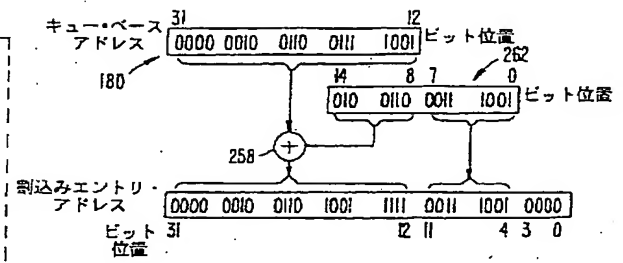
【図10】



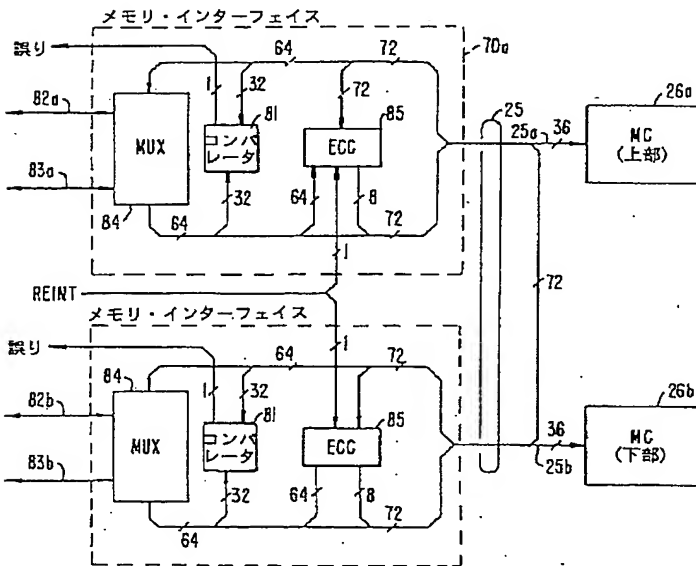
【図 1 3】



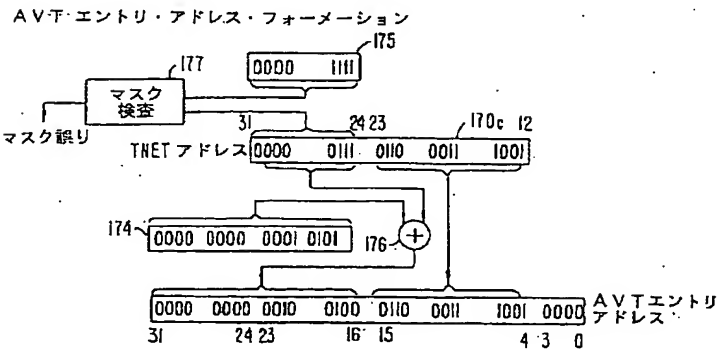
【図 2 2】



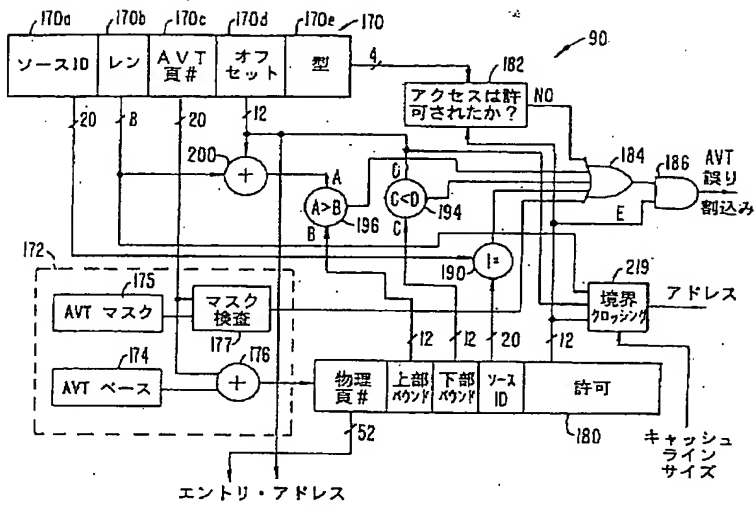
【図 1 5】



【図 1 7】



【図16】

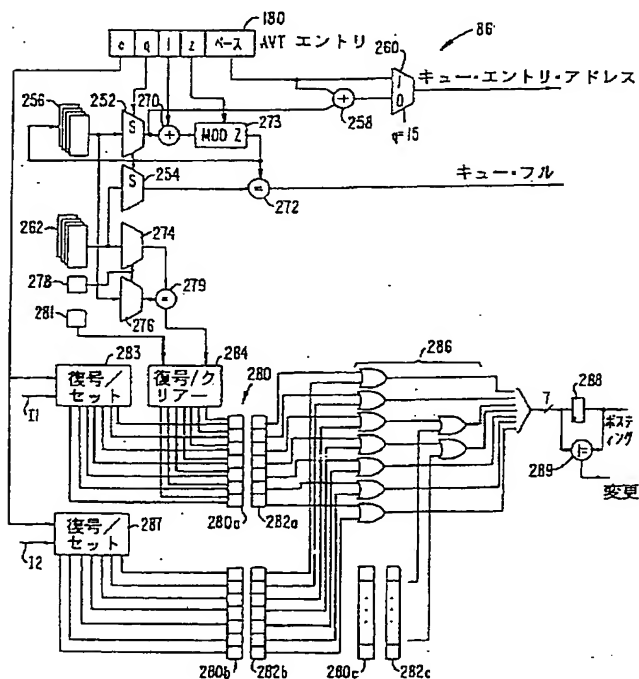


【図20】

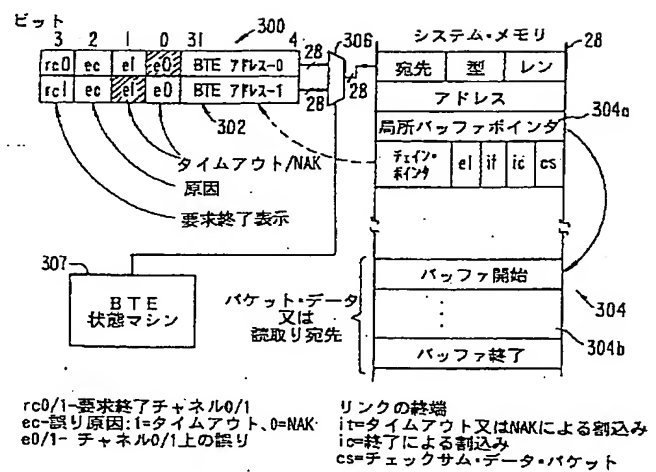
AVT エントリ (割込み)

64ビット	12ビット	20ビット	16ビット	16ビット
キューベース アドレス	許可	ソースID	(リザーブド)	c q l z

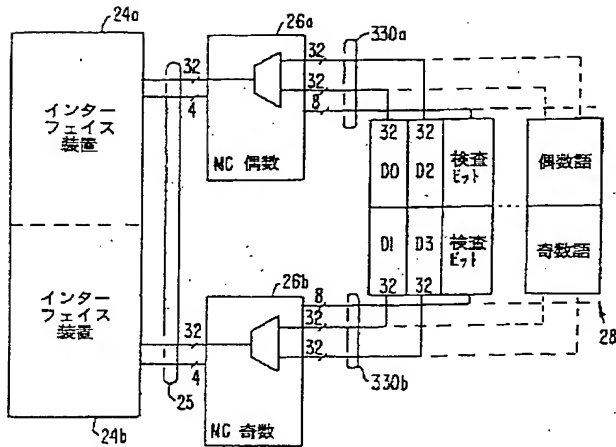
【図21】



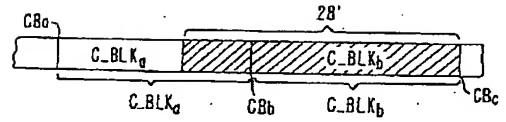
【図23】



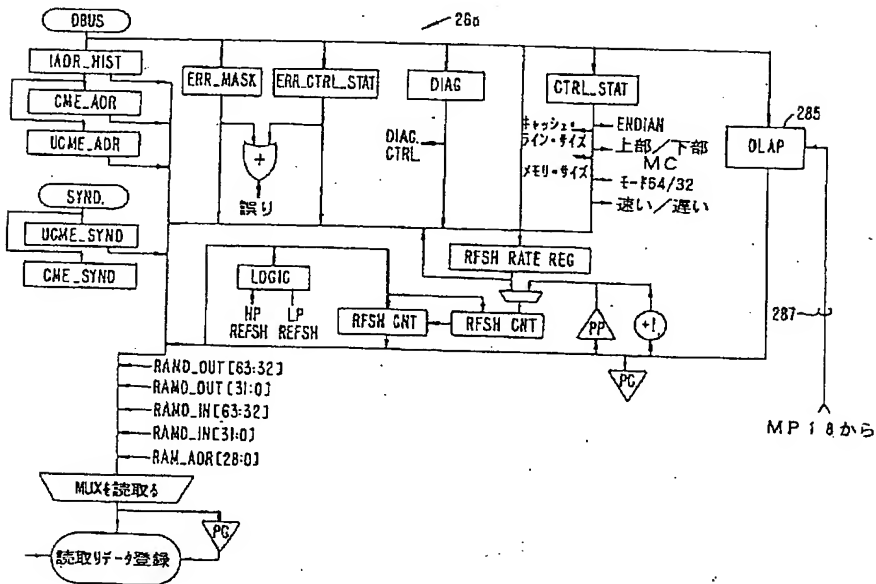
【図24】



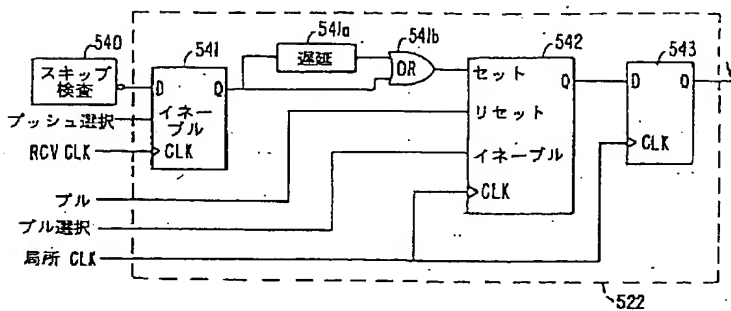
【図42】



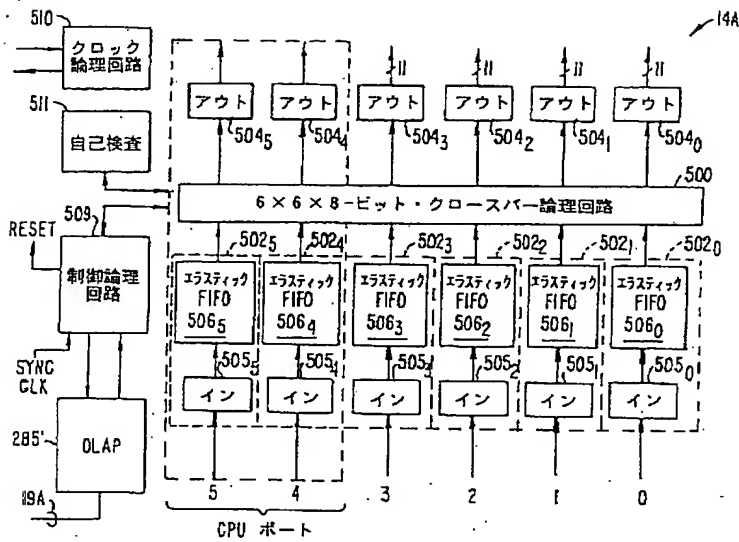
【図25】



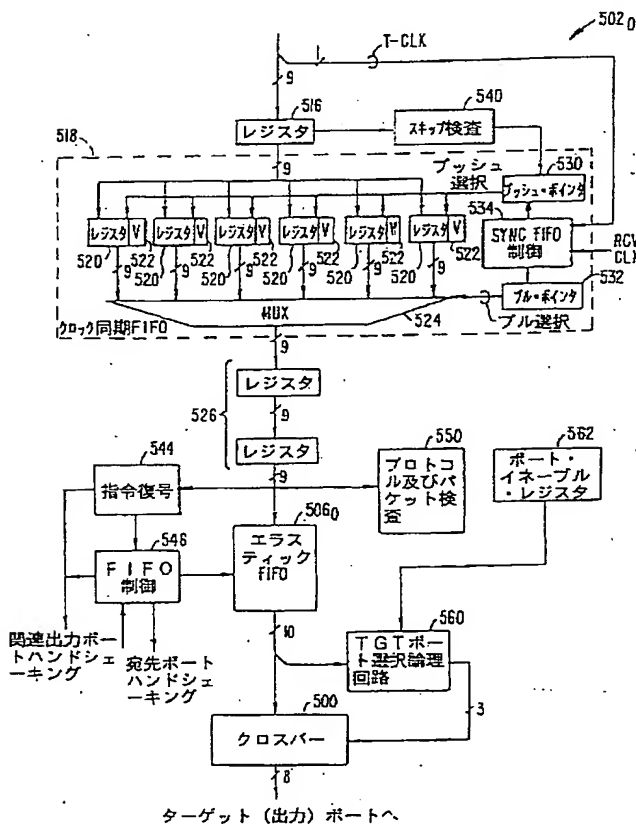
【図30】



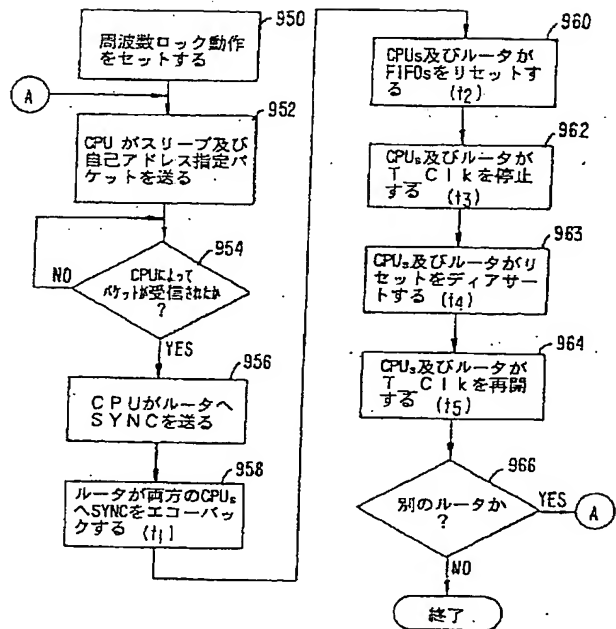
【図 27】



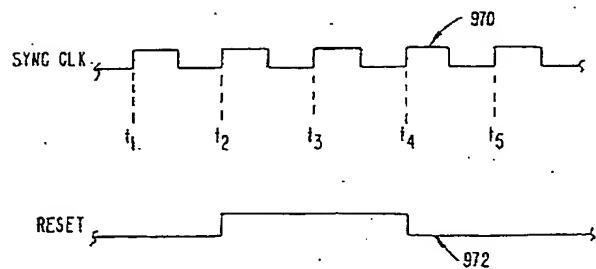
【図 29】



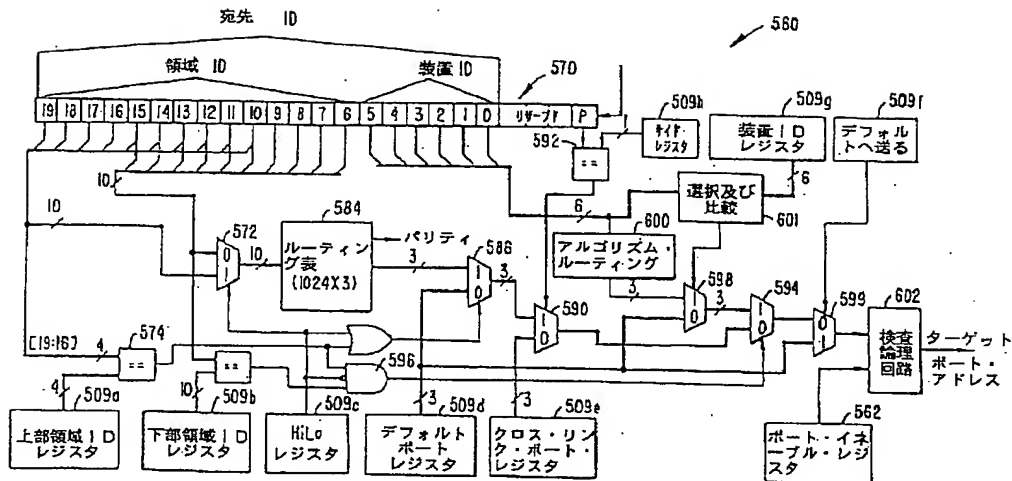
【図 45】



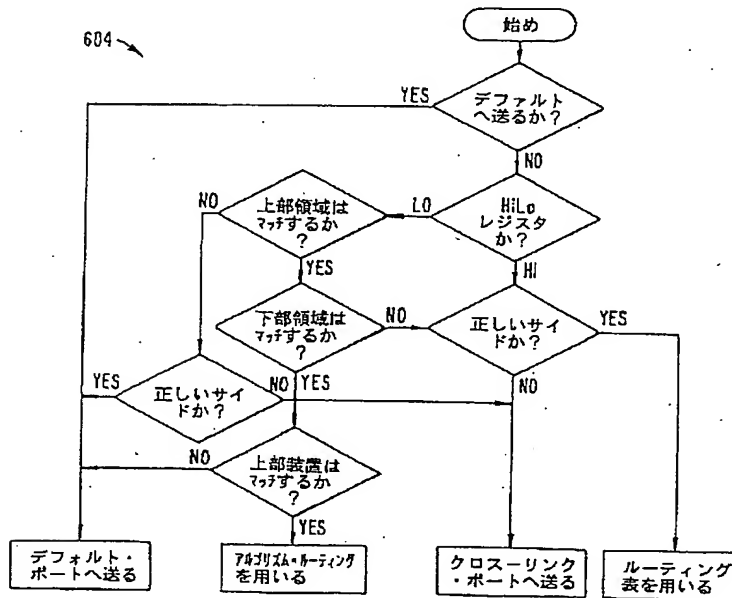
【図 46】



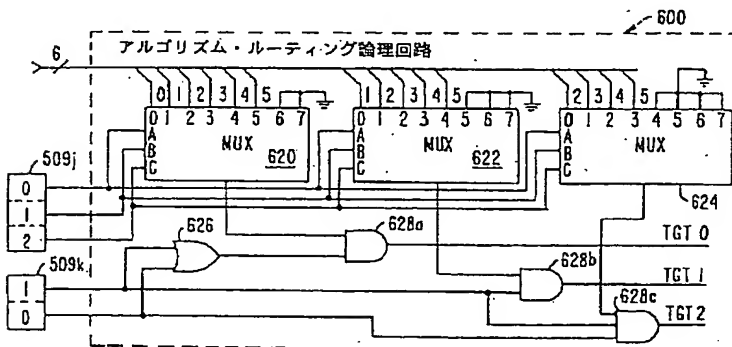
【図31】



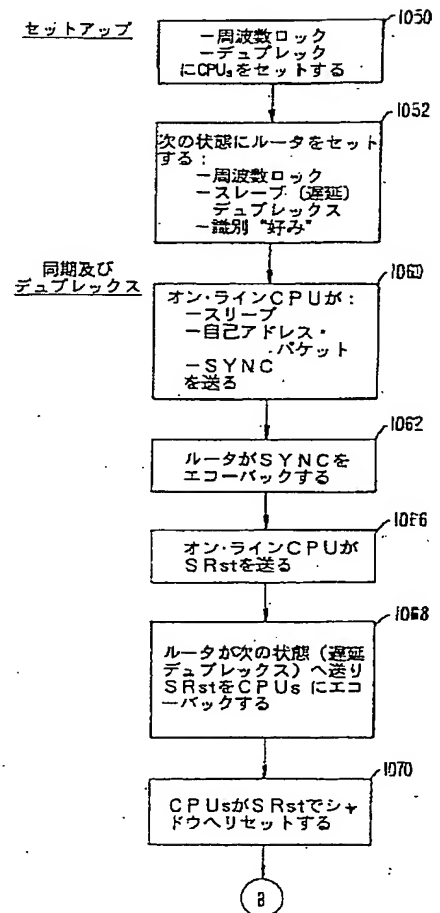
【図32】



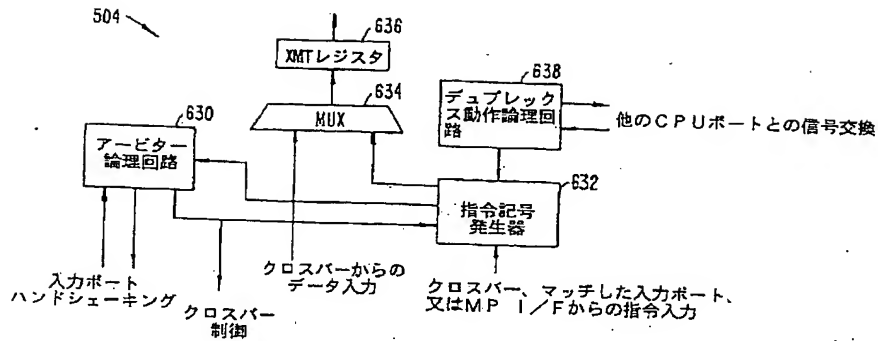
【図33】



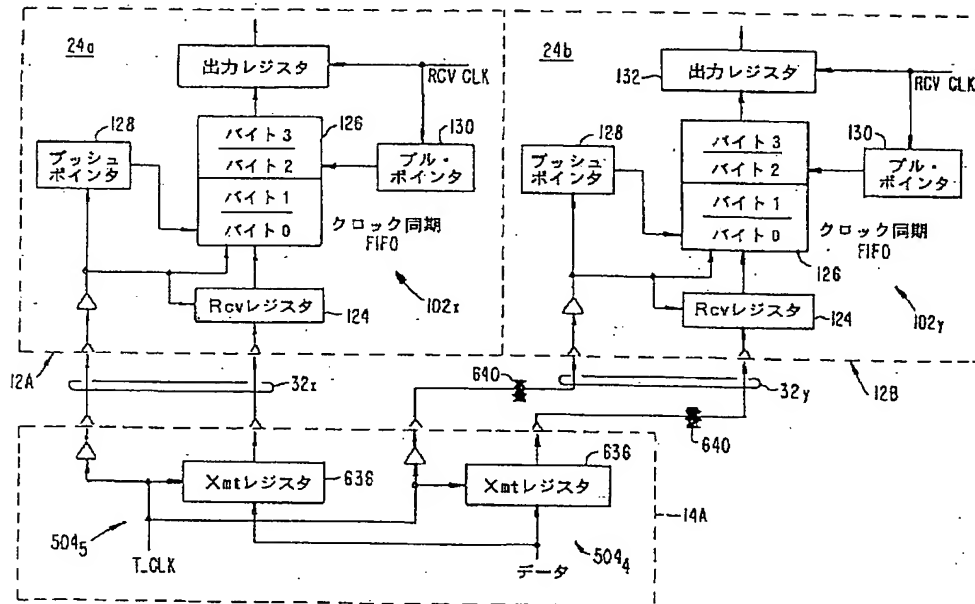
【図48】



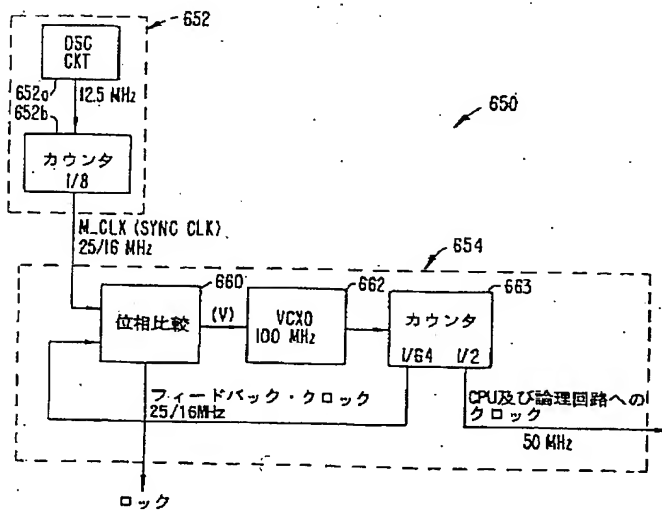
【図34】



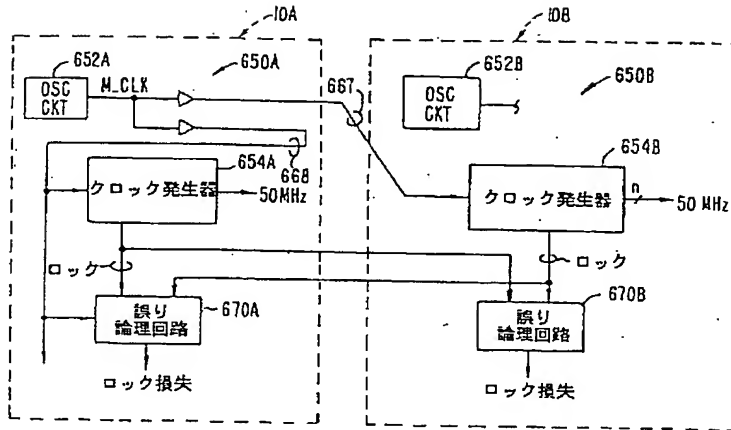
【図35】



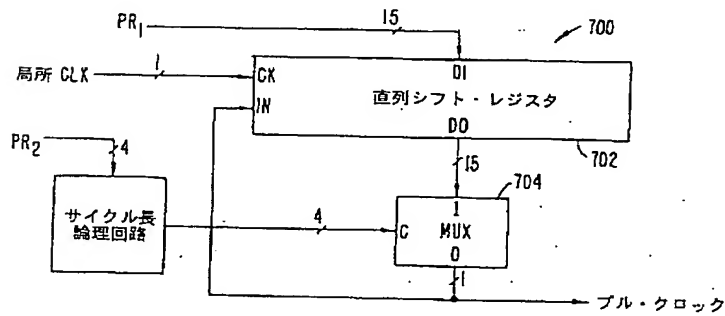
【図36】



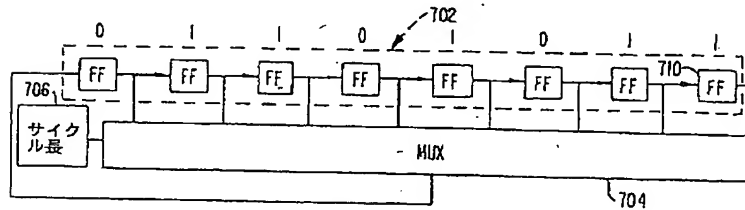
【図37】



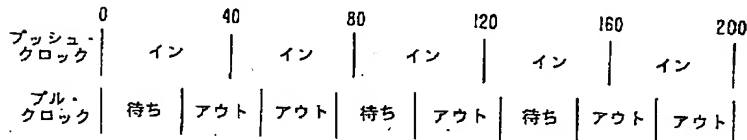
【図38】



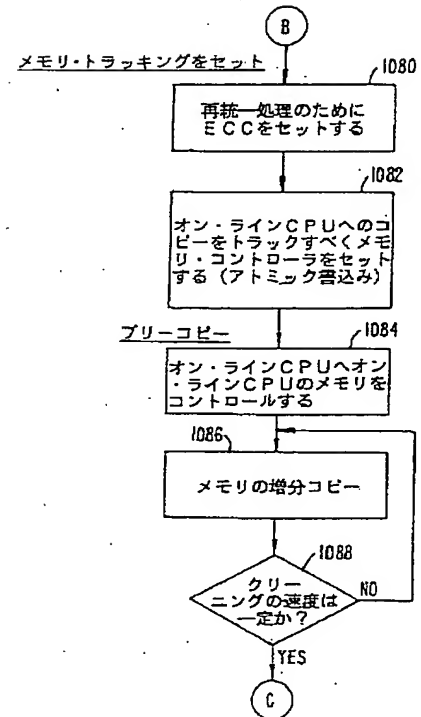
【図39】



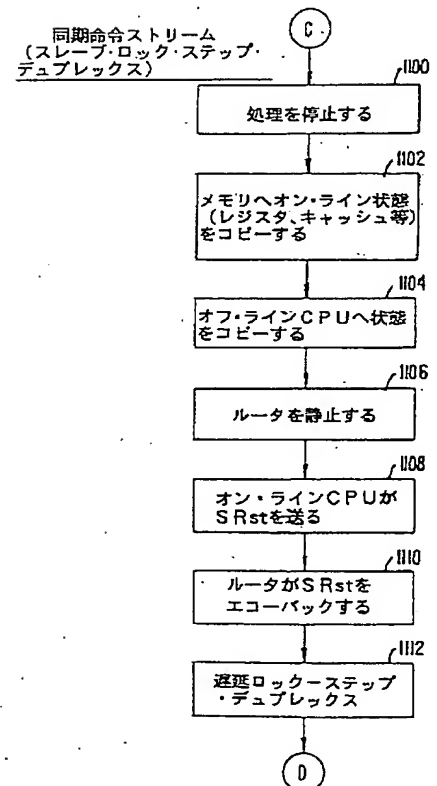
【図40】



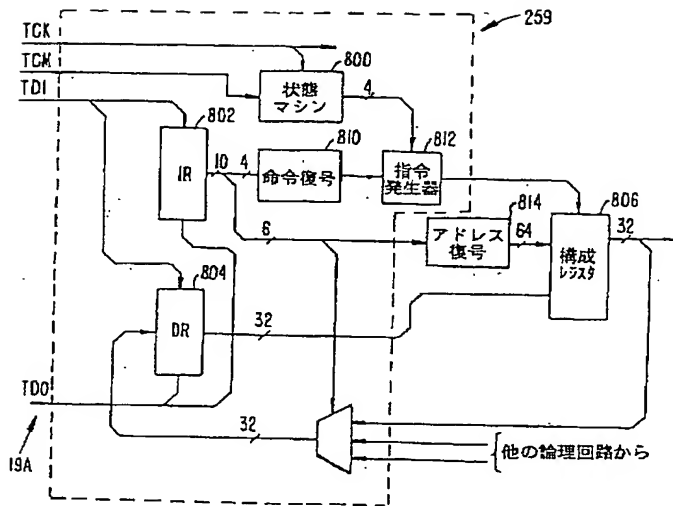
【図49】



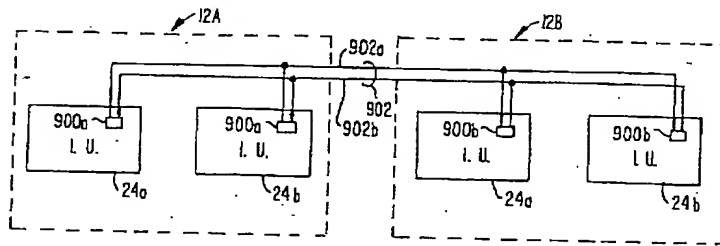
【図50】



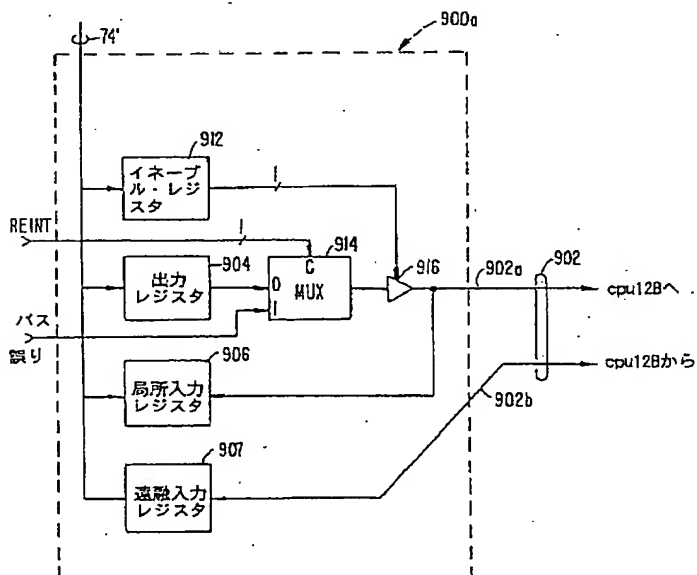
【図41】



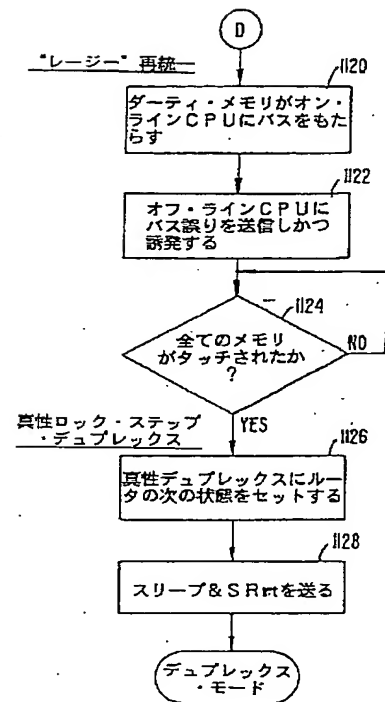
【図43】



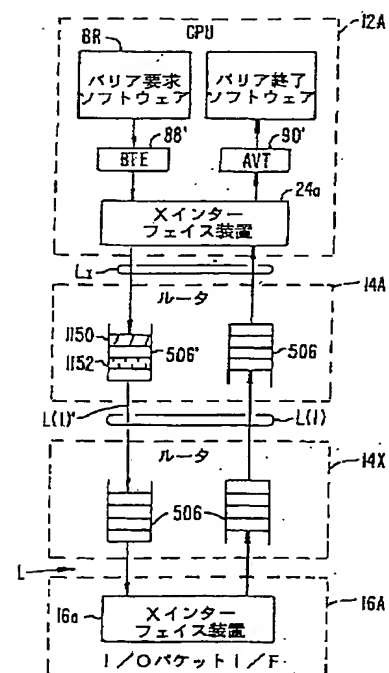
【図44】



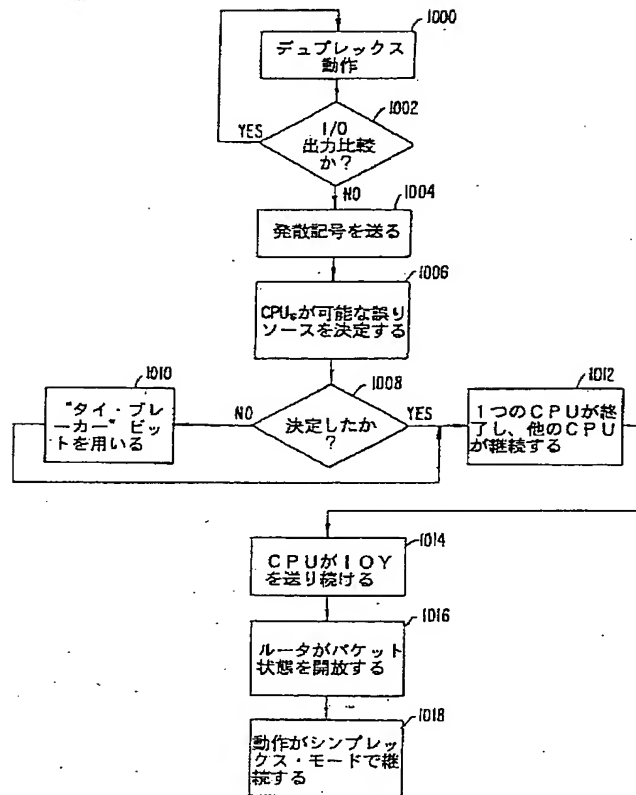
【図51】



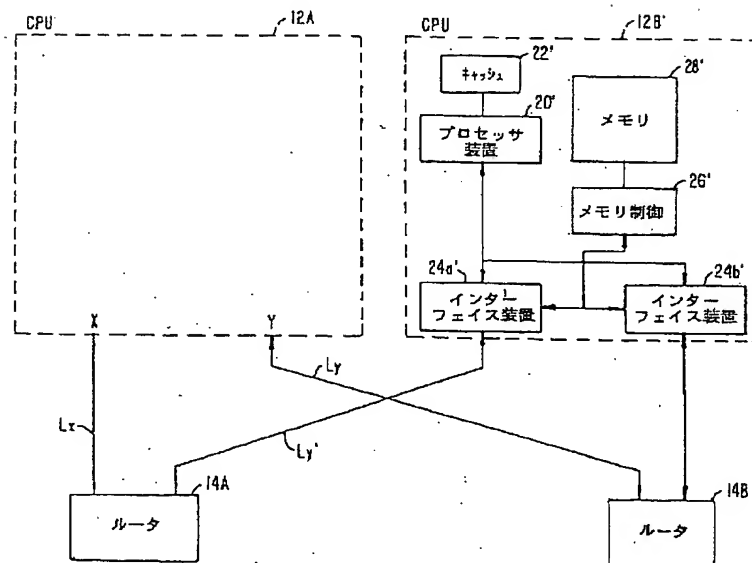
【図53】



【図 47】



【図 52】



フロントページの続き

(72)発明者 ディヴィッド ジェイ ガーシア
アメリカ合衆国 カリフォルニア州
95030 ロス ガトス ハッチンソン ロ
ード 24100

【公報種別】特許法第17条の2の規定による補正の掲載
【部門区分】第6部門第3区分
【発行日】平成15年9月10日(2003.9.10)

【公開番号】特開平9-128349
【公開日】平成9年5月16日(1997.5.16)
【年通号数】公開特許公報9-1284
【出願番号】特願平8-145550
【国際特許分類第7版】

G06F	15/16	360
		470
	11/18	310

【FI】

G06F	15/16	360 R
		470 J
	11/18	310 A

【手続補正書】

【提出日】平成15年6月6日(2003.6.6)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

【補正内容】

【特許請求の範囲】

【請求項1】 各々が複数のプロセッサを備え、Nが3以上の整数を表すN個の中央処理装置と、複数の入力／出力装置と、前記N個の中央処理装置のいずれか一つが前記入力／出力装置のいずれか一つへの別々でかつ独立した通信アクセスを有するように前記N個の中央処理装置の各々と前記入力／出力装置とを相互接続するネットワークとを備え、前記ネットワークは、前記N個の中央処理装置間でプロセッサ間通信を供給し、前記ネットワークは、前記中央処理装置の前記一つの内ロックステップ動作を可能にすべく前記中央処理装置の一つに対して少なくとも一つが通信を同期する、複数のルーティング装置を含み、それによって、前記入力／出力装置の一つを採り入れている前記N個の中央処理装置の故障したものの動作タス

クは、前記入力／出力装置の前記一つの採り入れを含んでいるその他のN個の中央処理装置のいずれかによって実行されることを特徴とする多重処理システム。

【請求項2】 第1の中央処理装置及び第2の中央処理装置を含んでおりかつNが2以上の整数である、各々が複数のプロセッサを備え、かつ各々が一つまたはそれ以上の処理を実行すべく動作する、N個の中央処理装置；前記第1の中央処理装置で実行される主要処理；前記第2の中央処理装置に関連付けられ、前記主要処理に動作及び機能において実質的に同一である、バックアップ処理；複数の入力／出力装置；前記N個の中央処理装置のいずれか一つが前記入力／出力装置のいずれか一つへの別々でかつ独立した通信アクセスを有するように前記N個の中央処理装置の各々と前記入力／出力装置とを相互接続するネットワークを備え、

前記ネットワークは、前記N個の中央処理装置間でプロセッサ間通信を供給し、前記ネットワークは、前記中央処理装置の前記一つの内ロックステップ動作を可能にすべく前記中央処理装置の一つに対して少なくとも一つが通信を同期する、複数のルーティング装置を含むことを特徴とする多重処理システム。